

# Hardness of Approximating the Minimum Distance of a Linear Code\*

Ilya Dumer<sup>†</sup>

Daniele Micciancio<sup>‡</sup>

Madhu Sudan<sup>§</sup>

April 15, 2007

## Abstract

We show that the minimum distance  $d$  of a linear code is not approximable to within any constant factor in random polynomial time (RP), unless NP (nondeterministic polynomial time) equals RP. We also show that the minimum distance is not approximable to within an additive error that is linear in the block length  $n$  of the code. Under the stronger assumption that NP is not contained in RQP (random quasi-polynomial time), we show that the minimum distance is not approximable to within the factor  $2^{\log^{1-\epsilon}(n)}$ , for any  $\epsilon > 0$ . Our results hold for codes over any finite field, including binary codes. In the process we show that it is hard to find approximately nearest codewords even if the number of errors exceeds the unique decoding radius  $d/2$  by only an arbitrarily small fraction  $\epsilon d$ . We also prove the hardness of the nearest codeword problem for asymptotically good codes, provided the number of errors exceeds  $(2/3)d$ .

Our results for the minimum distance problem strengthen (though using stronger assumptions) a previous result of Vardy who showed that the minimum distance cannot be computed *exactly* in deterministic polynomial time (P), unless  $P = NP$ . Our results are obtained by adapting proofs of analogous results for integer lattices due to Ajtai and Micciancio. A critical component in the adaptation is our use of linear codes that perform better than random (linear) codes.

## 1 Introduction

In this paper we study the computational complexity of two central problems from coding theory: (1) The complexity of approximating the minimum distance of a linear code and (2) The complexity of error-correction in codes of relatively large minimum distance. An error-correcting code  $\mathcal{A}$  of block length  $n$  over a  $q$ -ary alphabet  $\Sigma$  is a collection of strings (vectors) from  $\Sigma^n$ , 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  $q$  element. A code  $\mathcal{A} \subseteq \mathbb{F}_q^n$  is *linear* if it is closed under addition and

---

\*An edited version of this paper appears in *IEEE Transactions on Information Theory*, **49**(1):22-37, January 2003. This is the authors' copy.

<sup>†</sup>College of Engineering, University of California at Riverside. Riverside, CA 92521, USA. Email: [dumer@ee.ucr.edu](mailto:dumer@ee.ucr.edu). Research supported in part by NSF grant NCR-9703844.

<sup>‡</sup>Department of Computer Science and Engineering, University of California at San Diego. La Jolla, CA 92093-0114, USA. Email: [daniele@cs.ucsd.edu](mailto:daniele@cs.ucsd.edu). Research supported in part by NSF Career Award CCR-0093029.

<sup>§</sup>Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. 545 Technology Square, Cambridge, MA 02139, USA. Email: [madhu@mit.edu](mailto:madhu@mit.edu). Research supported in part by a Sloan Foundation Fellowship, an MIT-NEC Research Initiation Grant and NSF Career Award CCR-9875511.

multiplication by a scalar, i.e.,  $\mathcal{A}$  is a linear subspace of  $\mathbb{F}_q^n$  over 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<sup>1</sup>) is just its dimension as a vector space and the code can be compactly represented by a  $k \times n$  generator matrix  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  of rank  $k$  such that  $\mathcal{A} = \{\mathbf{x}\mathbf{A} \mid \mathbf{x} \in \mathbb{F}_q^k\}$ . An important property of a code is its *minimum distance*. For any vectors  $\mathbf{x}, \mathbf{y} \in \Sigma^n$ , the Hamming weight of  $\mathbf{x}$  is the number  $\text{wt}(\mathbf{x})$  of nonzero coordinates of  $\mathbf{x}$ . The weight function  $\text{wt}(\cdot)$  is a norm, and the induced metric  $d(\mathbf{x}, \mathbf{y}) = \text{wt}(\mathbf{x} - \mathbf{y})$  is called the Hamming distance. The (minimum) distance  $d(\mathcal{A})$  of the code  $\mathcal{A}$  is the minimum Hamming distance  $d(\mathbf{x}, \mathbf{y})$  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  $d(\mathcal{A})$  equals the weight  $\text{wt}(\mathbf{x})$  of the lightest nonzero codeword  $\mathbf{x} \in \mathcal{A} \setminus \{\mathbf{0}\}$ . If  $\mathcal{A}$  is a linear code over  $\mathbb{F}_q$  with block length  $n$ , rank  $k$  and minimum distance  $d$ , then it is customary to say that  $\mathcal{A}$  is a linear  $[n, k, d]_q$  code. Throughout the paper we use the following notational conventions: matrices are denoted by boldface uppercase Roman letters (e.g.,  $\mathbf{A}, \mathbf{C}$ ), the associated linear codes are denoted by the corresponding calligraphic letters (e.g.,  $\mathcal{A}, \mathcal{C}$ ), and we write  $\mathcal{A}[n, k, d]_q$  to mean that  $\mathcal{A}$  is a linear code over  $\mathbb{F}_q$  with block length  $n$ , information content  $k$  and minimum distance  $d$ .

## 1.1 The Minimum Distance Problem

Three of the four central parameters associated with a linear code, namely  $n$ ,  $k$  and  $q$ , are evident from its matrix representation. The minimum distance problem (MINDIST) is that of evaluating the fourth — namely — given a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  find the minimum distance  $d$  of the corresponding code  $\mathcal{A}$ . The minimum distance of a code is obviously related to its error correction capability  $\lfloor (d-1)/2 \rfloor$  and therefore finding  $d$  is a fundamental computational problem in coding theory. The problem gains even more significance in light of the fact that long  $q$ -ary codes chosen at random give the best parameters known for any  $q < 46$  (in particular, for  $q = 2$ )<sup>2</sup>. A polynomial time algorithm to compute the distance would be the ideal solution to the problem, as it could be used to construct good error correcting codes by choosing a generator matrix at random and checking if the associated code has a large minimum distance. Unfortunately, no such algorithm is known. The complexity of this problem (can it be solved in polynomial time or not?) was first explicitly questioned by Berlekamp, McEliece and van Tilborg [5] in 1978 who conjectured it to be NP-complete. This conjecture was finally resolved in the affirmative by Vardy [23, 22] in 1997, proving that the minimum distance cannot be computed in polynomial time unless  $P = NP$ . ([23, 22] also give further motivations and detailed account of prior work on this problem.)

To advance the search of good codes, one can relax the requirement of computing  $d$  exactly in two ways:

- Instead of requiring the exact value of  $d$ , one can allow for approximate solutions, i.e., an estimate  $d'$  that is guaranteed to be at least as big, but not much bigger than the true minimum distance  $d$ . (E.g.,  $d \leq d' \leq \gamma d$  for some approximation factor  $\gamma$ ).
- Instead of insisting on deterministic solutions that always produce correct (approximate) answers, one can consider randomized algorithms such that  $d' \leq \gamma d$  only holds in a probabilistic

---

<sup>1</sup>Throughout the paper, we write  $\log$  for the logarithm to the base 2, and  $\log_q$  when the base  $q$  is any number possibly different from 2.

<sup>2</sup>For square prime powers  $q \geq 49$ , linear AG codes can perform better than random ones [21] and are constructible in polynomial time. For all other  $q \geq 46$  it is still possible to do better than random codes, however the best known procedures to construct them run in exponential time [24].

sense. (Say  $d' \leq \gamma d$  with probability at least  $1/2$ .)<sup>3</sup>

Such algorithms can still be used to randomly generate relatively good codes as follows. Say we want a code with minimum distance  $d$ . We pick a generator matrix at random such that the code is expected to have minimum distance  $\gamma d$ . Then, we run the probabilistic distance approximation algorithm many times using independent coin tosses. If all estimates returned by the distance approximation algorithm are at least  $\gamma d$ , then the minimum distance of the code is at least  $d$  with very high probability.

In this paper, we study these more relaxed versions of the minimum distance problem and show that the minimum distance is hard to approximate (even in a probabilistic sense) to within any constant factor, unless  $\text{NP} = \text{RP}$  (i.e., every problem in NP has a polynomial time probabilistic algorithm that always rejects NO instances and accepts YES instances with high probability). Under the stronger assumption that NP does not have random quasi-polynomial<sup>4</sup> time algorithms (RQP), we prove that the minimum distance of a code of block length  $n$  is not approximable to within a factor of  $2^{\log^{(1-\epsilon)} n}$  for any constant  $\epsilon > 0$ . (This is a naturally occurring factor in the study of the approximability of optimization problems — see the survey of Arora and Lund [4].) Our methods adapt the proof of the inapproximability of the shortest lattice vector problem (SVP) due to Micciancio [18] (see also [19]) which in turn is based on Ajtai’s proof of the hardness of solving SVP [2].

## 1.2 The Error Correction Problem

In the process of obtaining the inapproximability result for the minimum distance problem, we also shed light on the general error-correction problem for linear codes. The simplest formulation is the Nearest Codeword Problem (NCP) (also known as the “maximum likelihood decoding problem”). Here, the input instance consists of a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  and a received word  $\mathbf{x} \in \mathbb{F}_q^n$  and the goal is to find the nearest codeword  $\mathbf{y} \in \mathcal{A}$  to  $\mathbf{x}$ . A more relaxed version is to estimate the minimum “error weight”  $d(\mathbf{x}, \mathcal{A})$  that is the distance  $d(\mathbf{x}, \mathbf{y})$  to the nearest codeword, without necessarily finding codeword  $\mathbf{y}$ . The NCP is a well-studied problem: Berlekamp, McEliece and van Tilborg [5] showed that it is NP-hard (even in its weight estimation version); and more recently Arora, Babai, Stern and Sweedyk [3] showed that the error weight is hard to approximate to within any constant factor unless  $\text{P} = \text{NP}$ , and within factor  $2^{\log^{(1-\epsilon)} n}$  for any  $\epsilon > 0$ , unless  $\text{NP} \subseteq \text{QP}$  (deterministic quasi-polynomial time). This latter result has been recently improved to inapproximability within  $2^{O(\log n / \log \log n)} = n^{1/O(\log \log n)}$  under the assumption that  $\text{P} \neq \text{NP}$  by Dinur, Kindler, Raz and Safra [9, 8]). On the positive side, NCP can be trivially approximated within a factor  $n$ . General, non-trivial approximation algorithms have been recently discovered by Berman and Karpinski [6], who showed that (for any finite field  $\mathbb{F}_q$ ) NCP can be approximated within a factor  $\epsilon n / \log n$  for any fixed  $\epsilon > 0$  in probabilistic polynomial time, and  $\epsilon n$  in deterministic polynomial time.

However the NCP only provides a first cut at understanding the error-correction problem. It shows that the error-correction problem is hard, if we try to decode *every linear code* regardless of the *error weight*. In contrast, the positive results from coding theory show how to perform error-correction in *specific linear codes* corrupted by *errors of small weight* (relative to the code

---

<sup>3</sup>One can also consider randomized algorithms with 2-sided error, where also the lower bound  $d' \geq d$  holds only in a probabilistic sense. All our results can be easily adapted to algorithms with 2-sided error.

<sup>4</sup> $f(n)$  is quasi-polynomial in  $n$  if it grows slower than  $2^{\log^c n}$  for some constant  $c$ .

distance). Thus the hardness of the NCP may come from one of two factors: (1) The problem attempts to decode every linear code and (2) The problem attempts to recover from too many errors. Both issues have been raised in the literature [23, 22], but only the former has seen some progress [7, 10, 20].

One problem that has been defined to study the latter phenomenon is the “Bounded distance decoding problem” (BDD, see [23, 22]). This is a special case of the NCP where the error weight is guaranteed (or “promised”) to be less than  $d(\mathcal{A})/2$ . This case is motivated by the fact that within such a distance, there may be at most one codeword and hence decoding is clearly unambiguous. Also this is the case where many of the classical error-correction algorithms (for say BCH codes, RS codes, AG codes, etc.) work in polynomial time.

### 1.3 Relatively Near Codeword Problem

To compare the general NCP, and the more specific BDD problem, we introduce a parameterized family of problems that we call the *Relatively Near Codeword Problem* (RNC). For real  $\rho$ ,  $\text{RNC}^{(\rho)}$  is the following problem:

Given a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  of a linear code  $\mathcal{A}$  of (not necessarily known) minimum distance  $d$ , an integer  $t$  with the promise that  $t < \rho \cdot d$ , and a received word  $\mathbf{x} \in \mathbb{F}_q^n$ , find a codeword within distance  $t$  from  $\mathbf{x}$ . (The algorithm may fail if the promise is violated, or if no such codeword exists. In other words, in contrast to the papers [3] and [5], the algorithm is expected to work only when the error weight is limited in proportion to the code distance.)

Both the nearest codeword problem (NCP) and the bounded distance decoding problem (BDD) are special cases of  $\text{RNC}^{(\rho)}$ :  $\text{NCP} = \text{RNC}^{(\infty)}$  while  $\text{BDD} = \text{RNC}^{(\frac{1}{2})}$ . Till recently, not much was known about  $\text{RNC}^{(\rho)}$  for constants  $\rho < \infty$ , leave alone  $\rho = \frac{1}{2}$  (i.e., the BDD problem). No finite upper bound on  $\rho$  can be easily derived from Arora et al.’s NP-hardness proof for NCP [3]. (In other words, their proof does not seem to hold for  $\text{RNC}^{(\rho)}$  for any  $\rho < \infty$ .) It turns out, as observed by Jain et al. [14], that Vardy’s proof of the NP-hardness of the minimum distance problem also shows the NP-hardness of  $\text{RNC}^{(\rho)}$  for  $\rho = 1$  (and actually extends to some  $\rho = 1 - o(1)$ ).

In this paper we significantly improve upon this situation, by showing NP-hardness (under randomized reductions) of  $\text{RNC}^{(\rho)}$  for every  $\rho > \frac{1}{2}$  bringing us much closer to an eventual (negative?) resolution of the bounded distance decoding problem.

### 1.4 Organization

The rest of the paper is organized as follows. In Section 2 we precisely define the coding problems studied in this paper, introduce some notation, and briefly overview the random reduction techniques and coding theory notions used in the rest of the paper. As explained in Section 2 our proofs rely on coding theoretic constructions that might be of independent interest, namely the construction of codes containing unusually dense clusters. These constructions constitute the main technical contribution of the paper, and are presented in Section 3 (for arbitrary linear codes) and Section 6 (for asymptotically good codes). The hardness of the relatively near codeword problem and the minimum distance problem are proved in Sections 4 and 5 respectively. Similar hardness results for asymptotically good codes are presented in Section 7. The reader mostly interested in

computational complexity may want to skip Section 3 at first reading, and jump directly to the NP-hardness results in Sections 4 and 5. Section 8 concludes with a discussion of the consequences and limitations of the proofs given in this paper, and related open problems.

## 2 Background

### 2.1 Approximation Problems

In order to study the computational complexity of coding problems, we formulate them in terms of promise problems. A *promise* problem is a generalization of the familiar notion of decision problem. The difference is that in a promise problem not every string is required to be either a YES or a NO instance. Given a string with the promise that it is either a YES or NO instance, one has to decide which of the two sets it belongs to.

The following promise problem captures the hardness of approximating the minimum distance problem within a factor  $\gamma$ .

**Definition 1 (Minimum Distance Problem)** *For prime power  $q$  and approximation factor  $\gamma \geq 1$ , an instance of  $\text{GAPDIST}_{\gamma,q}$  is a pair  $(\mathbf{A}, d)$ , where  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  and  $d \in \mathbb{Z}^+$ , such that*

- $(\mathbf{A}, d)$  is a YES instance if  $d(\mathcal{A}) \leq d$ .
- $(\mathbf{A}, d)$  is a NO instance if  $d(\mathcal{A}) > \gamma \cdot d$ .

In other words, given a code  $\mathbf{A}$  and an integer  $d$  with the promise that either  $d(\mathcal{A}) \leq d$  or  $d(\mathcal{A}) > \gamma \cdot d$ , one must decide which of the two cases holds true. The relation between approximating the minimum distance of  $\mathbf{A}$  and the above promise problem is easily explained. On the one hand, if one can compute a  $\gamma$ -approximation  $d' \in [d(\mathcal{A}), \gamma \cdot d(\mathcal{A})]$  to the minimum distance of the code, then one can easily solve the promise problem above by checking whether  $d' \leq \gamma \cdot d$  or  $d' > \gamma \cdot d$ . On the other hand, assume one has a decision oracle  $O$  that solves the promise problem above. Then, the minimum distance of a given code  $\mathbf{A}$  can be easily approximated using the oracle as follows. Notice that  $O(\mathbf{A}, n)$  always returns YES while  $O(\mathbf{A}, 0)$  always returns NO. Using binary search, one can efficiently find a number  $d$  such that  $O(\mathbf{A}, d) = \text{YES}$  and  $O(\mathbf{A}, d - 1) = \text{NO}$ .<sup>5</sup> This means that  $(\mathbf{A}, d)$  is *not* a NO instance and  $(\mathbf{A}, d - 1)$  is *not* a YES instance, and the minimum distance  $d(\mathcal{A})$  must lie in the interval  $[d, \gamma \cdot d]$ .

Similarly we can define the following promise problem to capture the hardness of approximating  $\text{RNC}^{(\rho)}$  within a factor  $\gamma$ .

**Definition 2 (Relatively Near Codeword Problem)** *For prime power  $q$ , and factors  $\rho > 0$  and  $\gamma \geq 1$ , an instance of  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  is a triple  $(\mathbf{A}, \mathbf{v}, t)$ , where  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$ ,  $\mathbf{v} \in \mathbb{F}_q^n$  and  $t \in \mathbb{Z}^+$ , such that  $t < \rho \cdot d(\mathcal{A})$  and<sup>6</sup>*

- $(\mathbf{A}, \mathbf{v}, t)$  is a YES instance if  $d(\mathbf{v}, \mathcal{A}) \leq t$ .
- $(\mathbf{A}, \mathbf{v}, t)$  is a NO instance if  $d(\mathbf{v}, \mathcal{A}) > \gamma t$ .

---

<sup>5</sup>By definition, the oracle can give any answer if the input is neither a YES instance nor a NO one. So, it would be wrong to conclude that  $(\mathbf{A}, d - 1)$  is a NO instance and  $(\mathbf{A}, d)$  is a YES one.

<sup>6</sup>Strictly speaking, the condition  $t < \rho \cdot d(\mathcal{C}_{\mathbf{A}})$  is a promise and hence should be added as a condition in both the YES and NO instances of the problem.

It is immediate that the problem  $\text{RNC}^{(\rho)}$  gets harder as  $\rho$  increases, since changing  $\rho$  has the only effect of weakening the promise  $t < \rho \cdot d(\mathcal{A})$ .  $\text{RNC}^{(\rho)}$  is the hardest when  $\rho = \infty$  in which case the promise  $t < \rho d(\mathcal{A})$  is vacuously true and we obtain the familiar (promise version of) the nearest codeword problem:

**Definition 3 (Nearest Codeword Problem)** For prime power  $q$  and  $\gamma \geq 1$ , an instance of  $\text{GAPNCP}_{\gamma,q}$  is a triple  $(\mathbf{A}, \mathbf{v}, t)$ ,  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$ ,  $\mathbf{v} \in \mathbb{F}_q^n$  and  $t \in \mathbb{Z}^+$ , such that

- $(\mathbf{A}, \mathbf{v}, t)$  is a YES instance if  $d(\mathbf{v}, \mathcal{A}) \leq t$ .
- $(\mathbf{A}, \mathbf{v}, t)$  is a NO instance if  $d(\mathbf{v}, \mathcal{A}) > \gamma \cdot t$ .

The promise problem  $\text{GAPNCP}_{\gamma,q}$  is NP-hard for every constant  $\gamma \geq 1$  (cf. [3]<sup>7</sup>), and this result is critical to our hardness result(s). We define one last promise problem to study the hardness of approximating the minimum distance of a code with linear additive error.

**Definition 4** For  $\tau > 0$  and prime power  $q$ , let  $\text{GAPDISTADD}_{\tau,q}$  be the promise problem with instances  $(\mathbf{A}, d)$ , where  $\mathbf{A} \in \mathbb{F}_q^{k \times n}$  and  $d \in \mathbb{Z}^+$ , such that

- $(\mathbf{A}, d)$  is a YES instance if  $d(\mathcal{A}) \leq d$
- $(\mathbf{A}, d)$  is a NO instance if  $d(\mathcal{A}) > d + \tau \cdot n$ .

## 2.2 Random Reductions and Techniques

The main result of this paper (see Theorem 22) is that approximation problem  $\text{GAPDIST}_{\gamma,q}$  is NP-hard for any constant factor  $\gamma \geq 1$  under polynomial *reverse unfaithful random* reductions (RUR-reductions, [15]), and for  $\gamma = 2^{\log^{(1-\epsilon)} n} = n^{1/\log^\epsilon(n)}$  it is hard under quasi-polynomial RUR-reductions. These are probabilistic reductions that map NO instances always to NO instances and YES instances to YES instances with high probability. In particular, given a security parameter  $s$ , all reductions presented in this paper warrant that YES instances are properly mapped with probability  $1 - q^{-s}$  in  $\text{poly}(s)$  time.<sup>8</sup> The existence of a (random) polynomial time algorithm to solve the hard problem would imply  $\text{NP} = \text{RP}$  (random polynomial time), i.e., every problem in NP would have a probabilistic polynomial time algorithm that always rejects NO instances and accepts YES instances with high probability.<sup>9</sup> Similarly, hardness for NP under quasi-polynomial RUR-reductions implies that the hard problem cannot be solved in RQP unless  $\text{NP} \subseteq \text{RQP}$  (random

<sup>7</sup>To be precise, Arora et al. [3] present the result only for binary codes. However, their proof is valid for any alphabet. An alternate way to obtain the result for any prime power is to use a recent result of Håstad [13] who states his result in linear algebra (rather than coding-theoretic) terms. We will state and use some of the additional features of the latter result in Section 7.

<sup>8</sup>Here, and in the rest of the paper, we use notation  $\text{poly}(n)$  to denote any polynomially bounded function of  $n$ , i.e., any function  $f(n)$  such that  $f(n) = O(n^c)$  for some constant  $c$  independent of  $n$ .

<sup>9</sup>Notice that we are unable to prove that the existence of a polynomial time approximation algorithm implies the stronger containment  $\text{NP} \subseteq \text{ZPP}$  (ZPP is the class of decision problems  $L$  such that both  $L$  and its complement are in RP, i.e.,  $\text{ZPP} = \text{RP} \cap \text{coRP}$ ), as done for example in [12]. The difference is that [12] proves hardness under *unfaithful random* reductions (UR-reductions [15], i.e. reductions that are always correct on YES instances and often correct on NO instances). Hardness under UR-reductions implies that no polynomial time algorithm exists unless  $\text{NP} \subseteq \text{coRP}$ , and therefore  $\text{RP} \subseteq \text{NP} \subseteq \text{coRP}$ . It immediately follows that  $\text{coRP} \subseteq \text{RP}$  and  $\text{NP} = \text{RP} = \text{coRP} = \text{ZPP}$ . However, in our case where RUR-reductions are used, we can only conclude that  $\text{RP} \subseteq \text{NP} \subseteq \text{RP}$ , and therefore  $\text{NP} = \text{RP}$ , but it is not clear how to establish any relation involving the complementary class  $\text{coRP}$  and  $\text{ZPP}$ .

quasi-polynomial time). Therefore, similarly to a proper NP-hardness result (obtained under deterministic polynomial reductions), hardness under polynomial RUR-reductions also gives evidence of the intractability of a problem.

In order to prove these results, we first study the problem  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$ . We show that the error weight is hard to approximate to within any constant factor  $\gamma$  and for any  $\rho > 1/2$  unless  $\text{NP} = \text{RP}$  (see Theorem 16). By using  $\gamma = 1/\rho$ , we immediately reduce our error-correction problem  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  to the minimum distance problem  $\text{GAPDIST}_{\gamma,q}$  for any constant  $\gamma < 2$ . We then use product constructions to “amplify” the constant and prove the claimed hardness results for the minimum distance problem.

The hardness of  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  for  $\rho > 1/2$  is obtained by adapting a technique of Micciancio [18], which is in turn based on the work of Ajtai [2] (henceforth Ajtai-Micciancio). They consider the analogous problem over the integers (rather than finite fields) with Hamming distance replaced by Euclidean distance. Much of the adaptation is straightforward; in fact, some of the proofs are even easier in our case due to the use of finite fields. The main hurdle turns out to be in adapting the following combinatorial problem considered and solved by Ajtai-Micciancio:

Given an integer  $k$  construct, in  $\text{poly}(k)$  time, integers  $r, l$ , an  $l$ -dimensional lattice  $\mathcal{L}$  (i.e., a subset of  $\mathbb{Z}^l$  closed under addition and multiplication by an integer) with minimum (Euclidean) distance  $d > r/\rho$  and a vector  $\mathbf{v} \in \mathbb{Z}^l$  such that the (Euclidean) ball of radius  $r$  around  $\mathbf{v}$  contains at least  $2^k$  vectors from  $\mathcal{L}$  (where  $\rho < 1$  is a constant independent of  $k$ ).

In our case we are faced with a similar problem with  $\mathbb{Z}^l$  replaced by  $\mathbb{F}_q^l$  and Euclidean distance replaced by Hamming distance. The Ajtai-Micciancio solution to the above problem involves number-theoretic methods and does not translate to our setting. Instead we show that if we consider a linear code whose performance (i.e., trade-off between rate and distance) is better than that of a random code, and pick a random light vector in  $\mathbb{F}_q^n$ , then the resulting construction has the required properties. We first solve this problem over sufficiently large alphabets using high rate Reed-Solomon (RS) codes. (See next subsection for the definition RS codes. This same construction has been used in the coding theory literature to demonstrate limitations to the “list-decodability” of RS codes [16].) We then translate the result to small alphabets using the well-known method of concatenating codes [11].

In the second part of the paper, we extend our methods to address asymptotically-good codes. We show that even for such codes, the Relatively Near Codeword problem is hard unless  $\text{NP}$  equals  $\text{RP}$  (see Theorem 31), though for these codes we are only able to prove the hardness of  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  for  $\rho > 2/3$ . Finally, we translate this to a result (see Theorem 32) showing that the minimum distance of a code is hard to approximate to within an additive error that is linear in the block length of the code.

## 2.3 Coding Theory

For a through introduction to coding theory the reader is referred to [17]. Here we briefly review some basic results as used in the rest of the paper. Readers with an adequate background in coding theory can safely skip to the next section.

A classical problem in coding theory is to determine for given alphabet  $\mathbb{F}_q$ , block length  $n$  and dimension  $k$ , what is the highest possible minimum distance  $d$  such that there exists a  $[l, m, d]_q$

linear code. The following two theorems give well known upper and lower bounds for  $d$ .

**Theorem 5 (Gilbert-Varshamov bound)** *For finite field  $\mathbb{F}_q$ , block length  $l$  and dimension  $m \leq l$ , there exists a linear code  $\mathcal{A}[l, m, d]_q$  with minimum distance  $d$  satisfying*

$$q^{m-l} \cdot |\mathcal{B}(\mathbf{0}, d-1)| \geq 1 \quad (1)$$

where  $|\mathcal{B}(\mathbf{0}, d-1)| = \sum_{r=0}^{d-1} \binom{l}{r} (q-1)^r$  is the volume of the  $l$  dimensional Hamming sphere of radius  $d-1$ . For any  $m \leq l$ , the smallest  $d$  such that (1) holds true is denoted  $d_m^*$ . (Notice that the definition of  $d_m^*$  depends both on the information content  $m$  and the block length  $l$ . For brevity, we omit  $l$  from the notation, as the block length is usually clear from the context.)

See [17] for a proof. The upper bound on  $d$  given by the Gilbert-Varshamov theorem (GV bound, for short) is not effective, i.e., even if the theorem guarantees the existence of linear codes with a certain minimum distance, the proof of the theorem employs a greedy algorithm that runs in exponential time, and we do not know any efficient way to find such codes for small alphabets  $q < 49$ . Interestingly, random linear codes meet the GV bound, i.e., if the generating matrix of the code is chosen uniformly at random, then the minimum distance of the resulting code satisfies the GV bound with high probability. However, given a randomly chosen generator matrix, it is not clear how to efficiently check whether the corresponding code meets the GV bound or not. We now give a lower bound on  $d$ .

**Theorem 6 (Singleton bound)** *For every linear code  $\mathcal{C}[l, m, d]_q$ , the minimum distance  $d$  is at most*

$$d \leq l - m + 1.$$

See [17] for a proof. A code  $\mathcal{C}[l, m, d]_q$  is called *Maximum Distance Separable* (MDS) if the Singleton bound is satisfied with equality, i.e., if  $d = l - m + 1$ . Reed-Solomon codes are an important example of MDS codes. For any finite field  $\mathbb{F}_q$ , and dimension  $m \leq q-1$ , the *Reed-Solomon* code (RS code)  $\mathcal{G}[q-1, m, q-m]_q$  can be defined as the set of  $(q-1)$ -dimensional vectors obtained evaluating all  $q^m$  polynomials  $p(x) = c_0 + c_1x + \dots + c_{m-1}x^{m-1} \in \mathbb{F}_q[x]$  of degree less than  $m$  at all nonzero points  $x \in \mathbb{F}_q \setminus \{0\}$ . *Extended Reed-Solomon* codes  $\mathcal{G}[q, m, q-m+1]_q$  are defined similarly, but evaluating the polynomials at all points  $x \in \mathbb{F}_q$ , including 0. Since a nonzero polynomial  $p$  of degree  $(m-1)$  can have at most  $(m-1)$  zeros, then every nonzero (extended) RS codeword has at most  $m-1$  zero positions, so it's Hamming weight is at least  $q-m$  (resp.  $q-m+1$ ). This proves that (extended) Reed-Solomon codes are maximum distance separable. Extended RS codes can be further extended considering the homogeneous polynomials  $p(x, y) = \sum_{i=1}^{m-1} a_i x^i y^{m-1-i}$ , and evaluating them at all points of the projective line  $\{(x, 1) : x \in \mathbb{F}_q\} \cup \{(1, 0)\}$ . This increases both the block length and the minimum distance by one, giving *twice extended* RS codes  $\mathcal{G}[q+1, m, q-m+2]_q$ .

Another family of codes we are going to use are the *Hadamard* codes  $\mathcal{H}[q^c-1, c, q^c-q^{c-1}]_q$ . In these codes, each codeword corresponds to a vector  $\mathbf{x} \in \mathbb{F}_q^c$ . The codeword associated to  $\mathbf{x}$  is obtained by evaluating all nonzero linear functions  $\phi : \mathbb{F}_q^c \rightarrow \mathbb{F}_q$  at  $\mathbf{x}$ . Since there are  $q^c$  linear functions in  $c$  variables (including the zero function), the code has block length  $q^c-1$ . Notice that any nonzero vector  $\mathbf{x} \in \mathbb{F}_q^c$  is mapped to 0 by exactly a  $1/q$  fraction of the linear functions  $\phi : \mathbb{F}_q^c \rightarrow \mathbb{F}_q$  (including the identically zero function). So the minimum distance of the code is  $q^c - q^{c-1}$ .

In Section 6 we will also use Algebraic Geometric codes (AG codes). The definition of these codes is beyond the scope of the present paper, and the interested reader is referred to [21].

An important operation used to combine codes is the concatenating code construction of [11]. Let  $\mathcal{A}$  and  $\mathcal{B}$  be two linear codes over alphabets  $\mathbb{F}_{q^k}$  and  $\mathbb{F}_q$ , with generating matrices  $\mathbf{A} \in \mathbb{F}_{q^k}^{m \times l}$  and  $\mathbf{B} \in \mathbb{F}_q^{n \times k}$ .  $\mathcal{A}$  is called the *outer* code, and  $\mathcal{B}$  is called the *inner* code. Notice that the dimension of the inner code equals the dimension of the alphabet of the outer code  $\mathbb{F}_{q^k}$ , viewed as a vector space over base field  $\mathbb{F}_q$ . The idea is to replace every component  $x_i$  of outer codeword  $[x_1, \dots, x_l] \in \mathcal{A} \subseteq \mathbb{F}_{q^k}^l$  with a corresponding inner codeword  $\phi(x_i) \in \mathcal{B}$ . More precisely, let  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_{q^k}$  be a basis for  $\mathbb{F}_{q^k}$  as a vector space over  $\mathbb{F}_q$  and let  $\phi: \mathbb{F}_{q^k} \rightarrow \mathcal{B}$  be the (unique) linear function such that  $\phi(\alpha_i) = \mathbf{b}_i$  for all  $i = 1, \dots, k$ . The concatenation function  $\diamond \mathcal{B}: \mathbb{F}_{q^k}^l \rightarrow \mathbb{F}_q^{ln}$  is given by

$$[x_1, \dots, x_n] \diamond \mathcal{B} = [\phi(x_1), \dots, \phi(x_n)].$$

Function  $\diamond \mathcal{B}$  is extended to sets of vectors in the usual way  $X \diamond \mathcal{B} = \{\mathbf{x} \diamond \mathcal{B}: \mathbf{x} \in X\}$ , and concatenated code  $\mathcal{A} \diamond \mathcal{B}$  is just the result of applying the concatenation function  $\diamond \mathcal{B}$  to set  $\mathcal{A}$ . It is easy to see that if  $\mathcal{A}$  is a  $[l, m, d]_{q^k}$  code and  $\mathcal{B}$  is a  $[n, k, t]_q$  code, then the concatenation  $\mathcal{A} \diamond \mathcal{B}$  is a  $[nl, km, dt]_q$  code with generator matrix  $\mathbf{C} \in \mathbb{F}_q^{nl \times km}$  given by

$$\mathbf{c}_{i+jk} = (\alpha_i \cdot \mathbf{a}_{j+1}) \diamond \mathcal{B}.$$

for all  $i = 1, \dots, k$  and  $j = 0, \dots, m - 1$ .

### 3 Dense Codes

In this section we present some general results about codes with a special density property (to be defined), and their algorithmic construction. The section culminates with the proof of Lemma 15 which shows how to efficiently construct a gadget that will be used in Section 4 in our NP-hardness proofs. Lemma 15 is in fact the only result from this section directly used in the rest of the paper (with the exception of Section 6 which extends the results of this section to asymptotically good codes). The reader mostly interested in computational complexity issues, may want to skip this section at first reading, and refer to Lemma 15 when used in the proofs.

#### 3.1 General overview

Let  $\mathcal{B}(\mathbf{v}, r) = \{\mathbf{x} \in \mathbb{F}_q^l \mid d(\mathbf{v}, \mathbf{x}) \leq r\}$  be the ball of radius  $r$  centered in  $\mathbf{v} \in \mathbb{F}_q^l$ . In this section, we wish to find codes  $\mathcal{C}[l, m, d]_q$  that include multiple codewords in some ball(s)  $\mathcal{B}(\mathbf{v}, r)$  of relatively small radius  $r = \lfloor \rho d \rfloor$ , where  $\rho$  is some positive real number. Obviously, the problem is meaningful only for  $\rho \geq 1/2$ , as any ball of radius  $r < d/2$  cannot contain more than a single codeword. Below we prove that for any  $\rho > 1/2$  it is actually possible to build such a code. These codes are used in the sequel to prove the hardness of  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  by reduction from the nearest codeword problem. Of particular interest is the case  $r < d$  (i.e.,  $\rho < 1$ ), as the corresponding hardness result for GAPRNC can be translated into an inapproximability result for the minimum distance problem.

We say that a code  $\mathcal{C}[l, m, d]_q$  is  $(\rho, k)$ -dense around  $\mathbf{v} \in \mathbb{F}_q^l$  if the ball  $\mathcal{B}(\mathbf{v}, \lfloor \rho d \rfloor)$  contains at least  $q^k$  codewords. We say that  $\mathcal{C}$  is  $(\rho, k)$ -dense if it is  $(\rho, k)$ -dense around  $\mathbf{v}$  for some  $\mathbf{v}$ . We want to determine for what values of the parameters  $\rho, k, l, m, d$  there exist  $(\rho, k)$ -dense codes. The technique we use is probabilistic: we show that there exist  $[l, m, d]_q$  codes such that the expected

number of codewords in a randomly chosen sphere  $\mathcal{B}(\mathbf{v}, \rho d)$  is at least  $q^k$ . It easily follows, by a simple averaging argument, that there exists a center  $\mathbf{v}$  such that the code is  $(\rho, k)$ -dense around  $\mathbf{v}$ . Let

$$\mu_{\mathcal{C}}(r) = \text{Exp}_{\mathbf{v} \in \mathbb{F}_q^l} [|\mathcal{C} \cap \mathcal{B}(\mathbf{v}, r)|] = \sum_{\mathbf{x} \in \mathcal{C}} \text{Pr}_{\mathbf{v} \in \mathbb{F}_q^l} \{\mathbf{x} \in \mathcal{B}(\mathbf{v}, r)\}$$

be the expected number of codewords in  $\mathcal{B}(\mathbf{v}, r)$  when the center  $\mathbf{v}$  is chosen uniformly at random from  $\mathbb{F}_q^l$ . Notice that

$$\mu_{\mathcal{C}}(r) = \sum_{\mathbf{x} \in \mathcal{C}} \text{Pr}_{\mathbf{v} \in \mathbb{F}_q^l} \{\mathbf{v} \in \mathcal{B}(\mathbf{x}, r)\} = \frac{|\mathcal{C}| \cdot |\mathcal{B}(\mathbf{0}, r)|}{q^l} = q^{m-l} \cdot |\mathcal{B}(\mathbf{0}, r)|. \quad (2)$$

This shows that the average number of codewords in a randomly chosen ball does not depend on the specific code  $\mathcal{C}$ , but only on its information content  $m$  (and the block length  $l$ , which is usually implicit and clear from the context). So, we can simply write  $\mu_m$  instead of  $\mu_{\mathcal{C}}$ .

In the rest of this subsection, we show how dense codes are closely related to codes that outperform the GV bound. This connection is not explicitly used in the rest of the paper, and it is presented here only for the purpose of illustrating the intuition behind the choice of codes used in the sequel. Using function  $\mu_m$ , the definition of the minimum distance (for codes with information content  $m$ ) guaranteed by the GV bound (see Theorem 5) can be rewritten as

$$d_m^* = \min\{r \mid \mu_m(r-1) \geq 1\}.$$

In particular, for any code  $\mathcal{C}[l, m, d]_q$ , the expected number of codewords in a random sphere of radius  $d_m^*$  is bigger than 1, and there must exist spheres with multiple codewords. If the distance of the code  $d$  exceeds the GV bound for codes with the same information content (i.e.,  $d > d_m^*$ ),  $\mathcal{C}$  is  $(\rho, k)$ -dense for some  $\rho = d_m^*/d < 1$  and  $k > 0$ . Several codes are known for which the minimum distance exceeds the GV bound:

- RS codes or, more generally, MDS codes, whose distances  $d$  exceed  $d_m^*$  for all code rates. RS codes of rate  $m/l$  approaching 1 will be of particular interest. This is due to the fact that the ratio  $d/d_m^*$  grows with code rate  $m/l$  of RS codes and tends to 2 for high rates. We implicitly use this fact in the sequel to build a  $\rho$ -dense codes for any  $\rho > 1/2$ .
- Binary BCH codes, whose distances exceed  $d_m^*$  for very high code rates approaching 1. These codes are not used in this paper, but they are an important family of codes that beat the GV bound.
- AG codes, meeting the TVZ bound, whose distances exceed  $d_m^*$  for most code rates bounded away from 0 and 1 for any  $q$ . These codes are used to prove hardness results for asymptotically good codes, and approximating the minimum distance up to an additive error.

**Remark 7** *The relation between dense codes and codes that exceed the GV bound can be given an exact, quantitative characterization. In particular, one can prove that for any positive integer  $k$  and real  $\rho$ , the expected number of codewords from  $\mathcal{C}[l, m, d]$  in a random sphere of radius  $r = \lfloor \rho d \rfloor$  is at least  $q^k$  if and only if  $(d_{m-k}^* - 1) \leq \rho d$ .*

### 3.2 Dense code families from MDS codes

We are interested in sequences of  $(\rho, k)$ -dense codes with fixed  $\rho$  and arbitrarily large  $k$ . Moreover, parameter  $k > 0$  should be *polynomially related* to the block length  $l$ , i.e.,  $l^\theta \leq k \leq l$  for some  $\theta \in (0, 1)$ . (Notice that  $k \leq l$  is a necessary condition because code  $\mathcal{C}$  contains at most  $q^l$  codewords and we want  $q^k$  distinct codewords in a ball.) This relation is essential to allow the construction of dense codes in time polynomial in  $k$ . Sequences of dense codes are formally defined below.

**Definition 8** *A sequence of codes  $\mathcal{C}_k[l_k, m_k, d_k]_{q_k}$  is called a  $\rho$ -dense code family if, for every  $k \geq 1$ ,  $\mathcal{C}_k$  is a  $(\rho, k)$ -dense code i.e., there exists a center  $\mathbf{v}_k \in \mathbb{F}_{q_k}^{l_k}$  such that the ball  $\mathcal{B}(\mathbf{v}_k, \lfloor \rho d_k \rfloor)$  contains  $q_k^k$  codewords. Moreover, we say that  $\mathcal{C}_k[l_k, m_k, d_k]_{q_k}$  is a polynomial code family if the block length  $l_k$  is polynomially bounded, i.e.,  $l_k \leq k^c$  for some constant  $c$  independent of  $k$ .*

In this section we use twice-extended RS codes to build  $\rho$ -dense code families for any  $\rho > 1/2$ .

**Lemma 9** *For any  $\epsilon \in (0, 1)$ , and sequence of prime powers  $q_k$  satisfying*

$$\lceil k/\epsilon \rceil^{1/\epsilon} \leq q_k \leq O(\text{poly}(k)), \quad (3)$$

*there exists a polynomial  $\rho$ -dense code family  $\mathcal{G}_k[l_k, m_k, d_k]_{q_k}$  with  $\rho = 1/(2(1 - \epsilon))$ .*

**Proof:** Fix the value of  $\epsilon$  and  $k$  and let  $\rho$  and  $q = q_k$  be as specified in the lemma. Define the radius  $r = \lfloor q^\epsilon \rfloor$ . Notice that, from the lower bound on the alphabet size, we get

$$\epsilon r = \epsilon \lfloor q^\epsilon \rfloor \geq \epsilon \lceil k/\epsilon \rceil \geq k. \quad (4)$$

In particular, since  $\epsilon < 1$ , we have  $r > k \geq 1$  and  $r < q$ . Since  $r$  is an integer, it must be

$$2 \leq r \leq q - 1.$$

Consider an MDS code  $\mathcal{G}[l, m, d]_q$  with  $l = q + 1$  and  $d = \lfloor r/\rho \rfloor + 1$  meeting the Singleton bound  $d = l - m + 1$  (e.g., twice-extended RS codes, see Section 2). From the definition of  $d$ , we immediately get

$$\rho d = \rho(\lfloor r/\rho \rfloor + 1) > r. \quad (5)$$

We will prove that  $\mu(r) \geq q^{\epsilon r}$ , i.e., the expected number of codewords in a randomly chosen ball of radius  $r$  is at least  $q^{\epsilon r}$ . It immediately follows from (4) and (5) that

$$\mu(\lfloor \rho d \rfloor) \geq \mu(r) \geq q^{\epsilon r} \geq q^k,$$

i.e., the code is  $(\rho, k)$ -dense on the average. Since MDS codes meet the Singleton bound with equality, we have

$$l - m = d - 1 = \lfloor r/\rho \rfloor \leq r/\rho = 2(1 - \epsilon)r.$$

Therefore the expected number of codewords in a random sphere satisfies

$$\mu_m(r) = q^{m-l} \cdot |\mathcal{B}(\mathbf{0}, r)| \geq q^{-2(1-\epsilon)r} \cdot |\mathcal{B}(\mathbf{0}, r)|. \quad (6)$$

We want to bound this quantity. Notice that for all  $l$  and  $r$  we have

$$\binom{l}{r} = \prod_{i=0}^{r-1} \frac{l-i}{r-i} \geq \left(\frac{l}{r}\right)^r. \quad (7)$$

where we have used that  $\frac{l-i}{r-i} \geq \frac{l}{r}$  for all  $r \leq l$  and  $i \in [0, r-1]$ . A slightly stronger inequality is obtained as follows:

$$\binom{l}{r} = \prod_{i=0}^{r-1} \frac{l-i}{r-i} \geq (l-r+1) \prod_{i=0}^{r-2} \frac{l}{r} = \left( \frac{r(l-r+1)}{l} \right) \left( \frac{l}{r} \right)^r \quad (8)$$

Moreover, the reader can easily verify that for all  $r \in [2, q]$

$$\frac{r(l-r+1)}{l} = \frac{r(q-r+2)}{q+1} \geq \frac{2q}{q+1}. \quad (9)$$

We use (8) and (9) to bound the volume of the sphere  $|\mathcal{B}(\mathbf{0}, r)|$  as follows:

$$\begin{aligned} |\mathcal{B}(\mathbf{0}, r)| &> \binom{l}{r} (q-1)^r \\ &\geq \left( \frac{2q}{q+1} \right) \left( \frac{q+1}{r} \right)^r (q-1)^r \\ &= \left( \frac{2q}{q+1} \right) \left( 1 - \frac{1}{q^2} \right)^r \left( \frac{q^2}{r} \right)^r \\ &\geq \left( \frac{2q}{q+1} \right) \left( 1 - \frac{r}{q^2} \right) q^{(2-\epsilon)r} \\ &\geq q^{(2-\epsilon)r} \end{aligned}$$

where in the last inequality we have used  $r \leq q-1$  and  $1 - (q-1)/q^2 \geq (q+1)/(2q)$ . Combining the bound  $|\mathcal{B}(\mathbf{0}, r)| > q^{(2-\epsilon)r}$  with inequality (6) we get

$$\mu_m(r) > q^{-2(1-\epsilon)r + (2-\epsilon)r} = q^{\epsilon r}.$$

This proves that our MDS codes are  $(\rho, k)$ -dense. Moreover, if  $q_k = O(\text{poly}(k))$ , then the block length  $l = q+1$  is also polynomial in  $k$ , and  $\mathcal{G}_k$  is a polynomial  $\rho$ -dense family of codes.  $\square$

### 3.3 Codes over a fixed alphabet

Lemma 9 gives a  $\rho$ -dense code family  $\mathcal{G}_k[l_k, m_k, d_k]_{q_k}$  for any fixed  $\rho > 1/2$  with  $q_k = O(\text{poly}(k))$ . In the sequel, we wish to find a  $\rho$ -dense family of codes over some fixed alphabet  $\mathbb{F}_q$ . In order to keep the alphabet size fixed and still get arbitrarily large  $k$ , we take the extension field  $\mathbb{F}_{q^c}$  and use the MDS codes from Lemma 9 with alphabet size  $q^c$ . These codes are concatenated with equidistant Hadamard codes  $\mathcal{H}[q^c-1, c, q^c - q^{c-1}]_q$  to obtain a family of dense codes over fixed alphabet  $\mathbb{F}_q$ . This procedure can be applied to any dense code as described in the following lemma.

**Lemma 10** *Let  $\mathcal{C}'[l', m', d']_{q^c}$  be an arbitrary code, and let  $\mathcal{H}[q^c-1, c, q^c - q^{c-1}]_q$  be the equidistant Hadamard code of size  $q^c$ . If  $\mathcal{C}'$  is  $(\rho, k)$ -dense (around some center  $\mathbf{v}$ ), then the concatenated code  $\mathcal{C} = \mathcal{C}' \diamond \mathcal{H}$  is  $(\rho, ck)$ -dense (around  $\mathbf{v} \diamond \mathcal{H}$ ).*

**Proof:** Let  $\mathcal{C}'[l', m', d']_{q^c}$  be a code and let  $\mathcal{H}[q^c-1, c, q^c - q^{c-1}]_q$  be the equidistant Hadamard code of size  $q^c$ . Define code  $\mathcal{C}$  as the concatenation of  $\mathcal{C}'$  and  $\mathcal{H}$ . (See Section 2 for details.) The resulting concatenated code  $\mathcal{C} = \mathcal{C}' \diamond \mathcal{H}$  has parameters

$$l = (q^c - 1)l', \quad m = cm', \quad d = (q^c - q^{c-1}) \cdot d'.$$

Now assume  $\mathcal{C}'$  is  $(\rho, k)$ -dense around some center  $\mathbf{v}$ . Notice that the concatenation function  $\mathbf{x} \mapsto \mathbf{x} \diamond \mathcal{H}$  is injective and satisfies  $\text{wt}(\mathbf{x} \diamond \mathcal{H}) = (d/d') \cdot \text{wt}(\mathbf{x})$ . Therefore the ball  $\mathcal{B}(\mathbf{x}, \rho d')$  is mapped into ball  $\mathcal{B}(\mathbf{v} \diamond \mathcal{H}, \rho d)$  and the number of  $\mathcal{C}'$ -codewords contained in  $\mathcal{B}(\mathbf{v}, \rho d')$  equals the number of  $(\mathcal{C}' \diamond \mathcal{H})$ -codewords contained in  $\mathcal{B}(\mathbf{v} \diamond \mathcal{H}, \rho d)$ . Therefore  $\mathcal{C}$  is  $(\rho, k')$  dense for  $k' = \log_q((q^c)^k) = ck$ .  $\square$

By increasing the degree  $c$  of the extension field  $\mathbb{F}_{q^c}$ , we obtain an infinite sequence of  $q$ -ary codes  $\mathcal{G} \diamond \mathcal{H}$ . Combining Lemma 9 and Lemma 10 we get the following proposition.

**Proposition 11** *For any  $\rho > 1/2$  and prime power  $q$ , there exists a polynomial  $\rho$ -dense family of codes  $\{\mathcal{A}_k\}_{k \geq 1}$  over a fixed alphabet  $\Sigma = \mathbb{F}_q$ .*

**Proof:** As in Lemma 9, let  $\epsilon > 0$  be an arbitrarily small constant and let  $\rho = 1/(2(1 - \epsilon))$ . For every  $k$ , define  $c_k = \lceil \frac{1}{\epsilon} \cdot \log_q \lceil \frac{k}{\epsilon} \rceil \rceil$ . Notice that  $q^{c_k}$  satisfies

$$\left\lceil \frac{k}{\epsilon} \right\rceil^{1/\epsilon} \leq q^{c_k} \leq O(\text{poly}(k)),$$

so we can invoke Lemma 9 with alphabet size  $q_k = q^{c_k}$  and obtain a  $(\rho, k)$ -dense code  $\mathcal{G}_k[l', m', d']_{q_k}$ . Let  $\mathcal{A}_k$  be the concatenation of  $\mathcal{G}_k$  with Hadamard code  $\mathcal{H}[q^{c_k} - 1, c_k, q^{c_k} - q^{c_k - 1}]_q$ . By Lemma 10,  $\mathcal{A}_k$  is  $(\rho, ck)$ -dense. Moreover, the block length of  $\mathcal{A}_k$  is  $l' \cdot (q^{c_k} - 1)$ , which is polynomial in  $k$ , because both  $l'$  and  $q^{c_k}$  are  $\text{poly}(k)$ . This proves that  $\mathcal{A}_k$  is a polynomial  $\rho$ -dense family.  $\square$

### 3.4 Polynomial construction

We proved that  $\rho$ -dense families of codes exist for any  $\rho > 1/2$ . In this subsection we address two issues related to the algorithmic construction of such codes:

- Can  $\rho$ -dense codes be constructed in polynomial time? I.e., is there an algorithm that on input  $k$  outputs (in time polynomial in  $k$ ) a linear code which is  $(\rho, k)$ -dense?
- Given a  $(\rho, k)$ -dense code, i.e., a code such that some ball  $\mathcal{B}(\mathbf{v}, \rho d)$  contains at least  $q^k$  codewords, can we efficiently find the center  $\mathbf{v}$  of a dense sphere?

The first question is easily answered: all constructions described in the previous subsection are polynomial in  $k$ , so the answer to the first question is yes. The second question is not as simple and to-date we do not know any deterministic procedure that efficiently produces dense codes together with a point around which the code is dense. However, we will see that, provided the code is dense “on the average”, the center of the sphere can be efficiently found at least in a probabilistic sense. We prove the following algorithmic variant of Proposition 11.

**Proposition 12** *For every prime power  $q$  and real constant  $\rho > 1/2$ , there exists a probabilistic algorithm that on input two integers  $k$  and  $s$ , outputs (in time polynomial in  $k$  and  $s$ ) three integers  $l, m, r$ , a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  and a center  $\mathbf{v} \in \mathbb{F}_q^l$  (of weight  $\text{wt}(\mathbf{v}) = r$ ) such that*

1.  $r < \rho \cdot d(\mathbf{A})$
2. with probability at least  $1 - q^{-s}$ , the ball  $\mathcal{B}(\mathbf{v}, r)$  contains  $q^k$  or more codewords.

Before proving the proposition, we make a few observations. The codes described in Proposition 11 are the concatenation of twice-extended RS codes with Hadamard codes, and therefore they can be explicitly constructed in polynomial time. While the codes are described explicitly, the proof that the code is dense is non constructive: in Lemma 9 we proved that the average number of codewords in a randomly chosen sphere is large, so spheres containing a large number of codewords certainly exist, but it is not clear how to find a dense center. At the first glance, one can try to reduce the entire set of centers  $\mathbb{F}_q^l$  used in Lemma 9. In particular, it can be proved that  $\mathbb{F}_q^l$  can be replaced by any MDS code  $\overline{\mathcal{G}}$  that includes our original MDS code  $\mathcal{G}$ . However, even with two centers left, we still need to count codewords in the two balls to find which is dense indeed. To-date, explicit procedures for finding a unique ball are yet unknown. Therefore below we use a probabilistic approach and show how to find a dense center with high probability.

First, let the center be chosen uniformly at random from the entire space  $\mathbb{F}_q^l$ . Given the expected number of codewords  $\mu$ , Markov's inequality yields

$$\Pr_{\mathbf{v} \in \mathbb{F}_q^l} (|\mathcal{B}(\mathbf{v}, r) \cap \mathcal{G}| > \Delta \cdot \mu) < 1/\Delta,$$

showing that spheres containing at most  $\Delta \cdot \mu$  codewords can be found with high probability  $1 - 1/\Delta$ . It turns out that a similar *lower* bound

$$\Pr_{\mathbf{v} \in \mathcal{B}(\mathbf{0}, r)} (|\mathcal{B}(\mathbf{v}, r) \cap \mathcal{G}| \leq \delta \cdot \mu) \leq \delta$$

can be proved if we restrict our choice of  $\mathbf{v}$  to a uniformly random element of  $\mathcal{B}(\mathbf{0}, r)$  only. This is an instance of a quite general lemma that holds for any pair of groups  $G \subseteq F$ .<sup>10</sup> In the lemma below, we use *multiplicative* notation for groups  $(G, \cdot)$  and  $(F, \cdot)$ . However, to avoid any possible confusion, we clarify in advance that in our application  $G$  and  $F$  will be groups  $(\mathcal{G}, +)$  and  $(\mathbb{F}_q^l, +)$  of codewords with respect to vector *sum* operation.

**Lemma 13** *Let  $F$  be a group,  $G \subset F$  a subgroup and  $B \subset F$  an arbitrary subset of  $F$ . Given  $z \in F$ , consider the subset  $Bz \stackrel{\text{def}}{=} \{b \cdot z \mid b \in B\}$  and let  $\mu$  be the average size of  $G \cap Bz$  as the element  $z$  runs through  $F$ . Choose  $v \in B^{-1} = \{b^{-1} \mid b \in B\}$  uniformly at random. Then for any  $\delta > 0$ ,*

$$\Pr_{v \in B^{-1}} \{|G \cap Bv| \leq \delta\mu\} \leq \delta.$$

**Proof:** Divide the elements of  $B$  into equivalence classes where  $u$  and  $v$  are equivalent if  $uv^{-1} \in G$ . Then choose  $v \in B^{-1} = \{b^{-1} \mid b \in B\}$  uniformly at random. If  $v = b^{-1}$  is chosen, then  $G \cap Bv$  has the same size as the equivalence class of  $b$ . (Notice: the equivalence class of  $b$  is  $Gb \cap B = (G \cap Bv)b$ .) Since the number of equivalence classes is (at most)  $|F|/|G|$ , the number of elements that belong to equivalence classes of size  $\delta\mu$  or less is bounded by  $\delta\mu|F|/|G|$  (i.e., the maximum number of classes times the maximum size of each class), and the probability that such a class is selected is at most  $\delta\mu|F|/(|G||B|)$ . The following simple calculation shows that  $\mu = |G||B|/|F|$  and therefore

---

<sup>10</sup>In fact, it is not necessary to have a group structure on the sets, and the lemma can be formulated in even more general settings, but working with groups make the presentation simpler.

the probability to select an element  $b$  such that  $|G \cap Bv| \leq \delta\mu$  is at most  $\delta$ :

$$\begin{aligned} \mu &= \text{Exp}_{z \in F} [|G \cap Bz|] \\ &= \sum_{y \in G} \text{Pr}_{z \in F} \{y \in Bz\} \\ &= \frac{|G| \cdot |B|}{|F|}. \end{aligned}$$

□

We are now ready to prove Proposition 12.

**Proof:** Fix some  $q$  and  $\rho > 1/2$  and let  $k, s$  be the input to the algorithm. We consider the  $(\rho, k')$ -dense code  $\mathcal{A}_{k'}$  from Proposition 11, where  $k' = k + s$ . This code is the concatenation of a twice extended RS code  $\mathcal{G}[l', m', d']_{q_{k'}}$  from Lemma 9, and a Hadamard code  $\mathcal{H}[q_{k'} - 1, c_{k'}, q_{k'}(1 - 1/q)]_q$  with block length polynomial in  $k'$ . Therefore a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  for  $\mathcal{A}_{k'}$  can be constructed in time polynomial in  $k, s$ .

At this point, we instantiate the Lemma 13 with groups  $F = (\mathbb{F}_{q'}^{l'}, +)$ ,  $G = (\mathcal{G}, +)$ , and  $B = \mathcal{B}(\mathbf{0}, r')$ , where  $r' = \lfloor \rho d' \rfloor$ . Notice that for any center  $z = \mathbf{z} \in \mathbb{F}_{q'}^{l'}$ , the set  $Bz$  is just the ball  $\mathcal{B}(\mathbf{z}, r')$  of radius  $r'$  centered in  $\mathbf{z}$ . From the proof of Lemma 9, the average size of  $G \cap Bz$  (i.e., the expected number of codewords in a random ball when the center is chosen uniformly at random from  $\mathbb{F}_{q'}^{l'}$ ) is at least  $q^{k+s}$ . Following Lemma 13, we choose  $\mathbf{v}' \in \mathbb{F}_{q'}^{l'}$  uniformly at random from  $\mathcal{B}(\mathbf{0}, r') = B = B^{-1}$ . By Lemma 13 (with  $\delta = q^{-s}$ ), we get that  $\mathcal{B}(\mathbf{v}', r')$  contains at least  $q^k$  codewords with probability  $1 - q^{-s}$ . Finally, by Lemma 10, we get that the corresponding ball  $\mathcal{B}(\mathbf{v}, r)$  in  $\mathbb{F}_q^l$  (with radius  $r = q_{k'}(1 - 1/q)r'$  and center  $\mathbf{v} = \mathbf{v}' \diamond \mathcal{H}$ ) contains at least  $q^k$  codewords from  $\mathcal{A}$ . The output of the algorithm is given by a generating matrix for code  $\mathcal{A}_{k'}$ , the block length  $l$  and information content  $m$  of this matrix, radius  $r$  and vector  $\mathbf{v}$ . □

### 3.5 Mapping dense balls onto full spaces

In the next section we will use the codewords inside the ball  $\mathcal{B}(\mathbf{v}, r)$  to represent the solutions to the nearest codeword problem. In order to be able to represent any possible solution, we need first to project the codewords in  $\mathcal{B}(\mathbf{v}, r)$  to the set of all strings over  $\mathbb{F}_q$  of some shorter length. This is accomplished in the next lemma by another probabilistic argument. Given a matrix  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$  and a vector  $\mathbf{y} \in \mathbb{F}_q^l$ , let  $\mathbf{T}(\mathbf{y}) = \mathbf{y}\mathbf{T}$  denote the linear transformation from  $\mathbb{F}_q^l$  to  $\mathbb{F}_q^k$ . Further, let  $\mathbf{T}(Y) = \{\mathbf{T}(\mathbf{y}) \mid \mathbf{y} \in Y\}$ .

**Lemma 14** *Let  $Y$  be any fixed subset of  $\mathbb{F}_q^l$  of size  $|Y| \geq q^{2k+s}$ . If matrix  $\mathbf{T} \in \mathbb{F}_q^{k \times l}$  is chosen uniformly at random, then with probability at least  $1 - q^{-s}$  we have  $\mathbf{T}(Y) = \mathbb{F}_q^k$ .*

**Proof:** Choose  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$  uniformly at random. We want to prove that with very high probability  $\mathbf{T}(Y) = \mathbb{F}_q^k$ . Choose a vector  $\mathbf{t} \in \mathbb{F}_q^k$  at random and define a new function  $\mathbf{T}'(\mathbf{y}) = \mathbf{y}\mathbf{T} + \mathbf{t}$ . Clearly  $\mathbf{T}'(Y) = \mathbb{F}_q^k$  if and only if  $\mathbf{T}(Y) = \mathbb{F}_q^k$ .

Notice that the random variables  $\mathbf{T}'(\mathbf{y})$  (indexed by vector  $\mathbf{y} \in Y$ , and defined by the random choice of  $\mathbf{T}$  and  $\mathbf{t}$ ) are pairwise independent and uniformly distributed. Therefore for any vector  $\mathbf{x} \in \mathbb{F}_q^k$ ,  $\mathbf{T}'(\mathbf{y}) = \mathbf{x}$  with probability  $p = q^{-k}$ . Let  $N_{\mathbf{x}}$  be the number of  $\mathbf{y} \in Y$  such that  $\mathbf{T}'(\mathbf{y}) = \mathbf{x}$ .

By linearity of expectation and pairwise independence of the  $\mathbf{T}'(\mathbf{y})$  we have  $\text{Exp}[N_{\mathbf{x}}] = |Y|p$  and  $\text{Var}[N_{\mathbf{x}}] = |Y|(p - p^2) < |Y|p$ . Applying Chebychev's inequality we get

$$\begin{aligned} \Pr\{N_{\mathbf{x}} = 0\} &\leq \Pr\{|N_{\mathbf{x}} - \text{Exp}[N_{\mathbf{x}}]| \geq \text{Exp}[N_{\mathbf{x}}]\} \\ &\leq \frac{\text{Var}[N_{\mathbf{x}}]}{\text{Exp}[N_{\mathbf{x}}]^2} \\ &< \frac{1}{|Y|p} \leq q^{-(k+s)}. \end{aligned}$$

Therefore, for any  $\mathbf{x} \in \mathbb{F}_q^k$ , the probability that  $\mathbf{T}'(\mathbf{y}) \neq \mathbf{x}$  for every  $\mathbf{y} \in Y$  is at most  $q^{-(k+s)}$ . By union bound, the probability that there exists some  $\mathbf{x} \in \mathbb{F}_q^k$  such that  $\mathbf{x} \notin \mathbf{T}'(Y)$  is at most  $q^{-s}$ , i.e., with probability at least  $1 - q^{-s}$ ,  $\mathbf{T}'(Y) = \mathbb{F}_q^k$  and therefore  $\mathbf{T}(Y) = \mathbb{F}_q^k$ .  $\square$

We combine Proposition 12 and Lemma 14 to build the gadget needed in the NP-hardness proofs in the following section.

**Lemma 15** *For any  $\rho > 1/2$  and finite field  $\mathbb{F}_q$  there exists a probabilistic polynomial time algorithm that on input  $k, s$  outputs, in time polynomial in  $k$  and  $s$ , integers  $l, m, r$ , matrices  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$ , and  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$  and a vector  $\mathbf{v} \in \mathbb{F}_q^l$  (of weight  $\text{wt}(\mathbf{v}) = r$ ) such that*

1.  $r < \rho \cdot d(\mathcal{A})$ .
2. with probability at least  $1 - q^{-s}$ ,  $\mathbf{T}(\mathcal{B}(\mathbf{v}, r) \cap \mathcal{A}) = \mathbb{F}_q^k$ , i.e., for every  $\mathbf{x} \in \mathbb{F}_q^k$  there exists a  $\mathbf{y} \in \mathcal{A}$  such that  $d(\mathbf{y}, \mathbf{v}) \leq r$  and  $\mathbf{y}\mathbf{T} = \mathbf{x}$ .

**Proof:** Run the algorithm of Proposition 12 on input  $k' = 2k + s + 1$  and  $s' = s + 1$  to obtain integers  $l, m, r$ , a matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  such that  $r < \rho d(\mathbf{A})$  and a center  $\mathbf{v} \in \mathbb{F}_q^l$  such that  $\mathcal{B}(\mathbf{v}, r)$  contains at least  $q^{2k+s+1}$  codewords with probability  $1 - q^{s+1}$ . Let  $Y$  be the set of all codewords in  $\mathcal{B}(\mathbf{v}, r)$ , and choose  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$  uniformly at random. By Lemma 14, the conditional probability, given  $|Y| \geq q^{2k+s+1}$ , that  $\mathbf{T}(Y) = \mathbb{F}_q^k$  is at least  $1 - q^{s+1}$ . Therefore, the probability that  $\mathbf{T}(Y) \neq \mathbb{F}_q^k$  is at most  $q^{s+1} + q^{s+1} \leq q^s$ .  $\square$

## 4 Hardness of the relatively near codeword problem

In this section we prove that the relatively near codeword problem is hard to approximate within any constant factor  $\gamma$  for all  $\rho > 1/2$ . The proof uses the gadget from Lemma 15.

**Theorem 16** *For any  $\rho' > 1/2$ ,  $\gamma' \geq 1$  and any finite field  $\mathbb{F}_q$ ,  $\text{GAPRNC}_{\gamma', q}^{(\rho')}$  is hard for NP under polynomial RUR-reductions. Moreover, the error probability can be made exponentially small in a security parameter  $s$  while maintaining the reduction polynomial in  $s$ .*

**Proof:** Fix some finite field  $\mathbb{F}_q$ . Let  $\rho$  be a real such that Lemma 15 holds true, let  $\epsilon > 0$  be an arbitrarily small positive real, and  $\gamma \geq 1$  such that  $\text{GAPNCP}_{\gamma, q}$  is NP-hard. We prove that  $\text{GAPRNC}_{\gamma', q}^{(\rho')}$  is hard for  $\rho' = \rho \cdot (1 + \epsilon)$  and  $\gamma' = \gamma/(2 + 1/\epsilon)$ . Since  $\epsilon$  can be arbitrarily small,

Lemma 15 holds for any  $\rho > 1/2$ , and  $\text{GAPNCP}_{\gamma,q}$  is NP-hard for any  $\gamma \geq 1$ , this proves the hardness of  $\text{GAPRNC}_{\gamma',q}^{(\rho')}$  for any  $\gamma' \geq 1$  and  $\rho' > 1/2$ .

The proof is by reduction from  $\text{GAPNCP}_{\gamma,q}$ . Let  $(\mathbf{C}, \mathbf{u}, t)$  be an instance of  $\text{GAPNCP}_{\gamma,q}$  with  $\mathbf{C} \in \mathbb{F}_q^{k \times n}$ . We want to define an instance  $(\mathbf{C}', \mathbf{u}', t')$  of  $\text{GAPRNC}_{\gamma',q}^{(\rho')}$  such that if  $(\mathbf{C}, \mathbf{u}, t)$  is a YES instance of  $\text{GAPNCP}_{\gamma,q}$ , then  $(\mathbf{C}', \mathbf{u}', t')$  is a YES instance of  $\text{GAPRNC}_{\gamma',q}^{(\rho')}$  with high probability, while if  $(\mathbf{C}, \mathbf{u}, t)$  is a NO instance of  $\text{GAPNCP}_{\gamma,q}$ , then  $(\mathbf{C}', \mathbf{u}', t')$  is a NO instance of  $\text{GAPRNC}_{\gamma',q}^{(\rho')}$  with probability 1. Notice that the main difference between the two problems is that while in  $(\mathbf{C}, \mathbf{u}, t)$  the minimum distance  $d(\mathcal{C})$  can be arbitrarily small, in  $(\mathbf{C}', \mathbf{u}', t')$  the minimum distance  $d(\mathcal{C}')$  must be relatively large (compared to error weight parameter  $t'$ ). The idea is to embed the original code  $\mathcal{C}$  in a higher dimensional space to make sure that the new code has large minimum distance. At the same time we want also to embed target vector  $\mathbf{u}$  in this higher dimensional space in such a way that the distance of the target from the code is roughly preserved. The embedding is easily performed using the gadget from Lemma 15. Details follow.

On input  $\text{GAPNCP}$  instance  $(\mathbf{C}, \mathbf{u}, t)$ , we invoke Lemma 15 on input  $k$  (the information content of input code  $\mathbf{C} \in \mathbb{F}_q^{k \times n}$ ) and security parameter  $s$ , to find integers  $l, m, r$ , a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$ , a mapping matrix  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$ , and a vector  $\mathbf{v} \in \mathbb{F}_q^l$  such that:

1.  $r < \rho \cdot d(\mathcal{A})$
2.  $\mathbf{T}(\mathcal{A} \cap \mathcal{B}(\mathbf{v}, r)) = \mathbb{F}_q^k$  with probability at least  $1 - q^{-s}$ .

Consider the linear code  $\mathbf{ATC} \in \mathbb{F}_q^{m \times n}$ . Notice that all  $m$  rows of matrix  $\mathbf{ATC}$  are codewords of  $\mathcal{C}$ . (However, only at most  $k$  are independent.) We define matrix  $\mathbf{C}'$  by concatenating<sup>11</sup>  $b = \lceil \frac{r}{t} \rceil$  copies of  $\mathbf{ATC}$  and  $a = \lceil \frac{bt}{er} \rceil$  copies of  $\mathbf{A}$ :

$$\mathbf{C}' = \underbrace{[\mathbf{A}, \dots, \mathbf{A}]_a}_{a} \underbrace{[\mathbf{ATC}, \dots, \mathbf{ATC}]_b}_{b} \quad (10)$$

and vector  $\mathbf{u}'$  as the concatenation of  $a$  copies of  $\mathbf{v}$  and  $b$  copies of  $\mathbf{u}$ :

$$\mathbf{u}' = \underbrace{[\mathbf{v}, \dots, \mathbf{v}]_a}_{a} \underbrace{[\mathbf{u}, \dots, \mathbf{u}]_b}_{b} \quad (11)$$

Finally, let  $t' = ar + bt$ . The output of the reduction is  $(\mathbf{C}', \mathbf{u}', t')$ .

Before we can prove that the reduction is correct, we need to bound the quantity  $\frac{ar}{bt}$ . Using the definition of  $a$  and  $b$  we get:

$$\frac{bt}{ar} \leq \frac{bt}{\left(\frac{bt}{er}\right)r} = \epsilon$$

and

$$\frac{ar}{bt} < \frac{\left(\frac{bt}{er} + 1\right)r}{bt} = \frac{1}{\epsilon} + \frac{r}{bt} \leq \frac{1}{\epsilon} + \frac{r}{\left(\frac{r}{t}\right)t} = \frac{1}{\epsilon} + 1.$$

---

<sup>11</sup>Here the word “concatenation” is used to describe the simple juxtaposition of matrices or vectors, and not the concatenating code construction of [11] used in Section 3.

So, we always have  $\frac{ar}{bt} \in [\frac{1}{\epsilon}, \frac{1}{\epsilon} + 1)$ . We can now prove the correctness of the reduction. In order to consider  $(\mathbf{C}', \mathbf{u}', t')$  as an instance of  $\text{GAPRNC}_{\gamma', q}^{(\rho')}$ , we first prove that  $t' < \rho' \cdot d(\mathcal{C}')$ . Indeed,  $d(\mathcal{C}') \geq a \cdot d(\mathcal{A}) > ar/\rho$  and therefore

$$\frac{t'}{d(\mathcal{C}')} < \frac{ar + bt}{ar/\rho} = \rho \left( 1 + \frac{bt}{ar} \right) \leq \rho(1 + \epsilon) = \rho'. \quad (12)$$

Now, assume  $(\mathbf{C}, \mathbf{u}, t)$  is a YES instance, i.e., there exists  $\mathbf{x}$  such that  $d(\mathbf{x}\mathbf{C}, \mathbf{u}) \leq t$ . Let  $\mathbf{y} = \mathbf{z}\mathbf{A}$  be a codeword in  $\mathcal{A}$  such that  $d(\mathbf{y}, \mathbf{v}) \leq r$  and  $\mathbf{y}\mathbf{T} = \mathbf{x}$ . We know such a codeword exists with probability at least  $1 - q^{-s}$ . In such a case, we have

$$d(\mathbf{z}\mathbf{C}', \mathbf{u}') = a \cdot d(\mathbf{z}\mathbf{A}, \mathbf{v}) + b \cdot d(\mathbf{z}\mathbf{ATC}, \mathbf{u}) \leq ar + bt = t' \quad (13)$$

proving that  $(\mathbf{C}', \mathbf{u}', t')$  is a YES instance.

Conversely, assume  $(\mathbf{C}, \mathbf{u}, t)$  is a NO instance, i.e., the distance of  $\mathbf{u}$  from  $\mathbf{C}$  is greater than  $\gamma t$ . We want to prove that for all  $\mathbf{z} \in \mathbb{F}_q^m$  we have  $d(\mathbf{z}\mathbf{C}', \mathbf{u}') > \gamma' t'$ . Indeed,

$$\begin{aligned} d(\mathbf{z}\mathbf{C}', \mathbf{u}') &\geq b \cdot d(\mathbf{z}(\mathbf{ATC}), \mathbf{u}) \\ &\geq b \cdot d(\mathcal{C}, \mathbf{u}) \\ &> b \cdot \gamma t \\ &= \gamma'(2 + 1/\epsilon)bt \\ &= \gamma'((1 + 1/\epsilon)bt + bt) \\ &> \gamma' \left( \left( \frac{ar}{bt} \right) bt + bt \right) \\ &= \gamma' t' \end{aligned} \quad (14)$$

proving that  $(\mathbf{C}', \mathbf{u}', t')$  is a NO instance. (Notice that NO instances get mapped to NO instances with probability 1, as required.)  $\square$

**Remark 17** *The reduction given here is a randomized many-one reduction (or a randomized Karp reduction) which fails with exponentially small probability. However it is not a Levin reduction: i.e., given a witness for a YES instance of the source of the reduction we do not know how to obtain a witness to YES instances of the target in polynomial time. The problem is that given a solution  $\mathbf{x}$  to the nearest codeword problem, one has to find a codeword  $\mathbf{y}$  in the sphere  $\mathcal{B}(\mathbf{v}, r)$  such that  $\mathbf{y}\mathbf{T} = \mathbf{x}$ . Our proof only asserts that with high probability such a codeword exists, but it is not known how to find it. This was the case also for the Ajtai-Micciancio hardness proof for the shortest vector problem, where the failure probability was only polynomially small.*

As discussed in subsection 2.2, hardness under polynomial RUR-reductions easily implies the following corollary.

**Corollary 18** *For any  $\rho > 1/2$ ,  $\gamma \geq 1$  and any finite field  $\mathbb{F}_q$ ,  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  is not in RP unless  $\text{NP} = \text{RP}$ .*

Since NP is widely believed to be different from RP, Corollary 18 gives evidence that no (probabilistic) polynomial time algorithm to solve  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  exists.

## 5 Hardness of the Minimum Distance Problem

In this section we prove the hardness of approximating the Minimum Distance Problem. We first derive an inapproximability result to within some constant bigger than one by reduction from  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$ . Then we use direct product constructions to amplify the inapproximability factor to any constant and to factors  $2^{\log^{(1-\epsilon)} n}$ , for any  $\epsilon > 0$ .

### 5.1 Inapproximability to within some constant

The inapproximability of  $\text{GAPDIST}_{\gamma,q}$  to within a constant  $\gamma \in (1, 2)$  immediately follows from the hardness of  $\text{GAPRNC}_{\gamma,q}^{(1/\gamma)}$ .

**Lemma 19** *For every  $\gamma \in (1, 2)$ , and every finite field  $\mathbb{F}_q$ ,  $\text{GAPDIST}_{\gamma,q}$  is hard for NP under polynomial RUR-reductions with exponentially small soundness error.*

**Proof:** The proof is by reduction from  $\text{GAPRNC}_{\gamma,q}^{\gamma^{-1}}$ . Let  $(\mathbf{C}, \mathbf{u}, t)$  be an instance of  $\text{GAPRNC}_{\gamma,q}^{\gamma^{-1}}$  with distance  $d(\mathcal{C}) > t/\rho = \gamma t$ . Assume without loss of generality that  $\mathbf{u}$  does not belong to code  $\mathcal{C}$ . (One can easily check whether  $\mathbf{u} \in \mathcal{C}$  by solving a system of linear equations. If  $\mathbf{u} \in \mathcal{C}$  then  $(\mathbf{C}, \mathbf{u}, t)$  is a YES instance because  $d(\mathbf{u}, \mathcal{C}) = 0$ , and the reduction can output some fixed YES instance of  $\text{GAPDIST}_{\gamma,q}$ .) Define the matrix

$$\mathbf{C}' = \begin{bmatrix} \mathbf{C} \\ \mathbf{u} \end{bmatrix}. \quad (15)$$

Assume  $(\mathbf{C}, \mathbf{u}, t)$  is a YES instance of  $\text{GAPRNC}_{\gamma,q}^{\gamma^{-1}}$ , i.e., there exists an  $\mathbf{x}$  such that  $d(\mathbf{x}\mathbf{C}, \mathbf{u}) \leq t$ . Then,  $(\mathbf{C}', t)$  is a YES instance of  $\text{GAPDIST}_{\gamma,q}$ , since nonzero vector  $\mathbf{x}\mathbf{C} - \mathbf{u}$  belongs to code  $\mathcal{C}'$  and has weight at most  $t$ .

Conversely, assume  $(\mathbf{C}, \mathbf{u}, t)$  is a NO instance of  $\text{GAPRNC}_{\gamma,q}^{\gamma^{-1}}$ . We prove that any nonzero vector  $\mathbf{y} = \mathbf{x}\mathbf{C} + \alpha\mathbf{u}$  of code  $\mathcal{C}'$  has weight above  $\gamma t$ . Indeed, if  $\alpha = 0$  then  $\mathbf{y}$  is a nonzero codeword of  $\mathcal{C}$  and therefore has weight  $\text{wt}(\mathbf{y}) > \gamma t$  (since  $d(\mathcal{C}) > \gamma t$ ). On the other hand, if  $\alpha \neq 0$  then  $\text{wt}(\mathbf{y}) = \text{wt}((\alpha^{-1}\mathbf{x})\mathbf{C} - \mathbf{u}) > \gamma t$  as  $d(\mathbf{u}, \mathcal{C}) > \gamma t$ . Hence,  $d(\mathcal{C}') > \gamma t$  and  $(\mathbf{C}', t)$  is a NO instance of  $\text{GAPDIST}_{\gamma,q}$ .  $\square$

### 5.2 Inapproximability to within bigger factors

To amplify the hardness result obtained above, we take the direct product of the code with itself. We first define direct products.

**Definition 20** *For  $i \in \{1, 2\}$ , let  $\mathcal{A}_i$  be a linear code generated by  $\mathbf{A}_i \in \mathbb{F}_q^{k_i \times n_i}$ . Then the direct product of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , denoted  $\mathcal{A}_1 \otimes \mathcal{A}_2$  is a code over  $\mathbb{F}_q$  of block length  $n_1 n_2$  and dimension  $k_1 k_2$ . Identifying the set  $\mathbb{F}_q^{n_1 n_2}$  of  $n_1 n_2$  dimensional vectors with the set  $\mathbb{F}_q^{n_2 \times n_1}$  of all  $n_2 \times n_1$  matrices in the obvious way, the direct product  $\mathcal{A}_1 \otimes \mathcal{A}_2$  is conveniently defined as the set of all matrices  $\{\mathbf{A}_2^T \mathbf{X} \mathbf{A}_1 \mid \mathbf{X} \in \mathbb{F}_q^{k_2 \times k_1}\}$  in  $\mathbb{F}_q^{n_2 \times n_1}$ .*

Notice that a generating matrix for the product code can be easily computed from  $\mathbf{A}_1$  and  $\mathbf{A}_2$ , defining a basis codeword  $(\mathbf{A}_1^{(i)})^T \mathbf{A}_2^{(j)}$  for every row  $\mathbf{A}_1^{(i)}$  of  $\mathbf{A}_1$  and row  $\mathbf{A}_2^{(j)}$  of  $\mathbf{A}_2$  (where  $\mathbf{x}^T$  denotes the transpose of vector  $\mathbf{x}$ , and  $\mathbf{x}^T \mathbf{y}$  is the standard “external” product of column vector  $\mathbf{x}^T$  and row vector  $\mathbf{y}$ ). Notice that the codewords of  $\mathcal{A}_1 \otimes \mathcal{A}_2$  are matrices whose rows are codewords

of  $\mathcal{A}_1$  and columns are codewords of  $\mathcal{A}_2$ . In our reduction we will need the following fundamental property of direct product codes. For completeness (see also [17]), we prove it below.

**Proposition 21** *For linear codes  $\mathcal{A}_1$  and  $\mathcal{A}_2$  of minimum distance  $d_1$  and  $d_2$ , their direct product is a linear code of distance  $d_1d_2$ .*

**Proof:** First,  $\mathbf{A}_2^T \mathbf{X} \mathbf{A}_1$  has at least  $d_1d_2$  nonzero entries if  $\mathbf{X} \neq \mathbf{0}$ . Indeed, consider the matrix  $\mathbf{X} \mathbf{A}_1$  whose rows are codewords from  $\mathcal{A}_1$ . Since this matrix is nonzero, some row is a nonzero codeword of weight  $d_1$  or more. Thus  $\mathbf{X} \mathbf{A}_1$  has at least  $d_1$  nonzero columns. Now consider the matrix  $\mathbf{A}_2^T (\mathbf{X} \mathbf{A}_1)$ . At least  $d_1$  columns of this matrix are nonzero codewords of  $\mathcal{A}_2$ . each of weight at least  $d_2$ , for a total weight of  $d_1d_2$  or more.

Second, we verify that the minimum distance of  $\mathcal{A}_1 \otimes \mathcal{A}_2$  is exactly  $d_1d_2$ . To see this consider vectors  $\mathbf{x}_i \in \mathbb{F}_q^{k_i}$  such that  $\mathbf{x}_i \mathbf{A}_i$  has exactly  $d_i$  nonzero elements. Then notice that the matrix  $M = \mathbf{A}_2^T \mathbf{x}_2^T \mathbf{x}_1 \mathbf{A}_1$  is a codeword of  $\mathcal{A}_1 \otimes \mathcal{A}_2$ . Expressing  $M$  as  $(\mathbf{x}_2 \mathbf{A}_2)^T (\mathbf{x}_1 \mathbf{A}_1)$  we see that its  $i$ -th column is zero if the  $i$ -th coordinate of  $\mathbf{x}_1 \mathbf{A}_1$  is zero and the  $j$ -th row of  $M$  is zero if the  $j$ th coordinate of  $\mathbf{x}_2 \mathbf{A}_2$  is zero. Thus  $M$  is zero on all but  $d_1$  columns and  $d_2$  rows and thus at most  $d_1d_2$  entries are nonzero.  $\square$

We can now prove the following theorem.

**Theorem 22** *For every finite field  $\mathbb{F}_q$  the following holds:*

- For every  $\gamma > 1$ ,  $\text{GAPDIST}_{\gamma,q}$  is hard for NP under polynomial RUR-reductions.
- For every  $\epsilon > 0$ ,  $\text{GAPDIST}_{\gamma,q}$  is hard for NP under quasi-polynomial RUR-reductions for  $\gamma(n) = 2^{\log^{(1-\epsilon)} n}$ .

*In both cases the error probability is exponentially small in a security parameter.*

**Proof:** Let  $\gamma_0$  be such that  $\text{GAPDIST}_{\gamma_0,q}$  is hard by Lemma 19. Given an instance  $(\mathbf{A}, d)$  of  $\text{GAPDIST}_{\gamma_0,q}$ , consider the instance  $(\mathbf{A}^{\otimes l}, d^l)$  of  $\text{GAPDIST}_{\gamma_0,q}$ , where

$$\mathbf{A}^{\otimes l} = \underbrace{(\dots ((\mathbf{A} \otimes \mathbf{A}) \otimes \mathbf{A}) \dots \otimes \mathbf{A})}_l$$

is a generator matrix of

$$\mathcal{A}^{\otimes l} = \underbrace{(\dots ((\mathcal{A} \otimes \mathcal{A}) \otimes \mathcal{A}) \dots \otimes \mathcal{A})}_l$$

for an integer parameter  $l \in \mathbb{Z}^+$ . By Proposition 21 it follows that YES instances map to YES instances and NO instances to NO instances. Setting  $l = \frac{\log \gamma}{\log \gamma_0}$  yields the first part of the theorem. Notice that for constant  $l$ , the size of  $\mathbf{A}^{\otimes l}$  is polynomial in  $\mathbf{A}$ , and  $\mathbf{A}^{\otimes l}$  can be constructed in polynomial time (for any fixed  $l$  independent of the size of  $\mathbf{A}$ ).

To show the second part, we use the first part by setting  $\gamma_0 = 2$  and  $l = \log^{\frac{1-\epsilon}{\epsilon}} n$  in the previous reduction, where  $n$  is the block length of  $\mathcal{A}$ . This time the block length of  $\mathbf{A}^{\otimes l}$  will be  $N = n^l = 2^{\log^{1/\epsilon} n}$  which is quasi-polynomial in the block length  $n$  of the original instance  $\mathbf{A}$ . The reduction can be computed in quasi-polynomial (in  $n$ ) time, and the approximation factor achieved is

$$\gamma(N) = 2^l = 2^{\log^{\frac{1-\epsilon}{\epsilon}} n} = 2^{\log^{1-\epsilon} N}.$$

□

As for the relatively near codeword problem, the following corollary can be easily derived from the hardness result under RUR-reductions.

**Corollary 23** *For every finite field  $\mathbb{F}_q$  the following holds:*

- For every  $\gamma > 1$ ,  $\text{GAPDIST}_{\gamma,q}$  is not in RP unless  $\text{NP} = \text{RP}$ .
- For every  $\epsilon > 0$ ,  $\text{GAPDIST}_{\gamma,q}$  is not in RQP for  $\gamma(n) = 2^{\log^{(1-\epsilon)} n}$  unless  $\text{NP} \subseteq \text{RQP}$ .

## 6 Asymptotically good dense codes

Our concatenated codes used in the proof of Proposition 11 employ outer MDS codes over  $\mathbb{F}_{q^c}$  and inner Hadamard codes  $\mathcal{H}$ , for fixed  $q$  and growing  $c$ . As a result, overall code rate vanishes for growing lengths, since so does the inner rate  $c/q^c$ . Relative distance also tends to 0, both in outer MDS codes and overall concatenations. To date, all other codes used to prove NP-hardness also have been asymptotically bad. To get hardness results (in both the relatively near codeword problem and the minimum distance problem) even when the codes are asymptotically good, we will need constructions of asymptotically good “dense” codes. To get such constructions, we will fix the outer alphabet  $q^c$  and the inner code  $\mathcal{H}$ . Then we use algebraic-geometry (AG) codes over  $\mathbb{F}_{q^c}$  to show that there exist dense families of asymptotically good codes over  $\mathbb{F}_q$ . In particular, we show that there exist  $\rho$ -dense code families  $\mathcal{C}_k[l_k, m_k, d_k]_q$  with relative distance  $d_k/l_k \geq \delta$  and rate  $m_k/l_k \geq R$ , where  $\rho < 1$ ,  $\delta > 0$  and  $R > 0$  are positive real numbers<sup>12</sup> independent of  $k$ . These codes will be used to prove the NP-hardness of  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  restricted to asymptotically good codes, and the hardness of approximating the minimum distance of a code within an additive linear error.

Given a square prime power  $q \geq 49$ , we use long algebraic-geometry codes  $\mathcal{A}[l, m, d]_q$  meeting the Tsfasman-Vlăduț-Zink (TVZ) bound  $d \geq l - m - \frac{l}{\sqrt{q-1}}$  (see [21]). The generator matrices  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  of these codes can be constructed for any  $l$  and  $m \leq l$  in time polynomial in  $l$ . As remarked in Section 3, the TVZ bound exceeds the GV bound for most code rates and therefore allows to obtain sequences of dense codes with fixed  $q$ . The resulting code sequence turns out to be  $\rho$ -dense for  $\rho > 2/3$ . We don’t know how to get  $\rho$  arbitrarily close to  $1/2$  using this technique.

**Proposition 24** *For every prime power  $q$  and every  $\rho > 2/3$  there exist  $R > 0$  and  $\delta > 0$  such that there is a polynomial family of asymptotically good  $\rho$ -dense codes  $\mathcal{C}_k[l_k, m_k, d_k]_q$  with rate  $m_k/l_k \geq R$  and relative distance  $d_k/l_k \geq \delta$ .*

**Proof:** Let  $\epsilon \in (0, 1/2]$  and  $\rho = \frac{2}{3(1-2\epsilon)}$ . Notice that by taking  $\epsilon$  sufficiently close to 0, we can get  $\rho$  arbitrarily close to  $2/3$ . We first prove the result assuming  $q \geq (2/\epsilon)^{2/\epsilon}$  and that  $q$  is a square. In this case the algebraic-geometry codes (for appropriate choice of parameters) already give what we want. Later we will use concatenation to get the proposition for arbitrary  $q$ .

---

<sup>12</sup>Notice that the relative distance satisfies  $d/l > (r/\rho)/l \geq \delta/\rho$  and the code is asymptotically good. In fact, for all dense codes considered in this paper, we have  $\rho < 1$  and therefore the relative distance is at least  $\delta$ .

**Case 1** ( $q \geq (2/\epsilon)^{2/\epsilon}$  and  $q$  is a square): Given parameter  $k$ , Let  $r = \lceil 2k/\epsilon \rceil$ ,  $l = 2k(\sqrt{q} - 1)$  and  $m = \lceil l - (\frac{3}{2} - \epsilon)r \rceil$ . Let  $\mathcal{A}[l, m, d]_q$  be a code meeting the TVZ bound, i.e., with  $d \geq l - m - \frac{l}{\sqrt{q}-1}$ . We show that this code has rate  $R \geq \frac{3}{4}$ , relative distance  $\delta \geq \frac{1}{2(\sqrt{q}-1)}$ ,  $r/d \leq \rho$ , and that the expected number of codewords of  $\mathcal{A}$  in a random ball of radius  $r$  is at least  $q^k$ . We start with bounding the rate:

$$\begin{aligned} \frac{m}{l} &\geq \frac{l - (\frac{3}{2} - \epsilon)r}{l} \\ &\geq 1 - \frac{3r}{2l} \\ &= 1 - \frac{3\lceil 2k/\epsilon \rceil}{4k(\sqrt{q} - 1)} \\ &\geq 1 - \frac{3}{4} \cdot \frac{(2/\epsilon) + 1}{\sqrt{q} - 1}. \end{aligned}$$

We want to prove that this bound is strictly positive. Using the assumption  $\epsilon \leq 1/2$  and  $q \geq (2/\epsilon)^{2/\epsilon}$  we get

$$\begin{aligned} \frac{(2/\epsilon) + 1}{\sqrt{q} - 1} &\leq \frac{(2/\epsilon) + 1}{(2/\epsilon)^{1/\epsilon} - 1} \\ &\leq \frac{(2/\epsilon) + 1}{(2/\epsilon)^2 - 1} \\ &= \frac{1}{(2/\epsilon) - 1} \\ &\leq \frac{1}{3}. \end{aligned}$$

Substituting in the bound for the rate of the code we get

$$\frac{m}{l} \geq 1 - \frac{3}{4} \cdot \frac{1}{3} = \frac{3}{4}.$$

Next we bound the minimum distance. From the TVZ bound we get

$$d \geq l - m - \frac{l}{\sqrt{q}-1} = \left\lfloor \left( \frac{3}{2} - \epsilon \right) r \right\rfloor - 2k \geq \left( \frac{3}{2} - \epsilon \right) r - 3k. \quad (16)$$

Using  $\epsilon \leq 1/2$  and dividing by the block length of the code, we get

$$\frac{d}{l} \geq \frac{r - 3k}{l} \geq \frac{2k/\epsilon - 3k}{2k(\sqrt{q} - 1)} \geq \frac{1}{2(\sqrt{q} - 1)}.$$

This proves that the relative distance is bounded away from 0, and the code is asymptotically good. We still need to prove that the code is  $(\rho, k)$ -dense (on the average), i.e.,  $r < \rho d$  and the expected number of codewords in a random ball of radius  $r$  is at least  $q^k$ .

We use (16) to bound the ratio  $d/r$ .

$$\begin{aligned}
\frac{d}{r} &\geq \frac{3}{2} - \epsilon - \frac{3k}{r} \\
&\geq \frac{3}{2} - \epsilon - \frac{3k}{2k/\epsilon} \\
&= \frac{3}{2} - \frac{5}{2}\epsilon \\
&> \frac{3}{2}(1 - 2\epsilon) = \frac{1}{\rho}.
\end{aligned}$$

This proves that the radius  $r$  is small, relative to the minimum distance of the code. We want to bound  $\mu_m(r)$ , the expected number of codewords in a random ball of radius  $r$ . Using (2) and (7), we have

$$\begin{aligned}
\mu_m(r) &\geq \binom{l}{r} (q-1)^r q^{m-l} \\
&\geq \left(\frac{l}{r}\right)^r (q-1)^r q^{-(\frac{3}{2}-\epsilon)r} \\
&\geq \left(\frac{2k(\sqrt{q}-1)}{\frac{2k}{\epsilon}+1}\right)^r \left(\frac{1}{\sqrt{q}}\right)^r \left(1-\frac{1}{q}\right)^r q^{\epsilon r} \\
&\geq \left(\frac{\epsilon}{1+1/4}\right)^r \left(1-\frac{1}{\sqrt{q}}\right)^r \left(1-\frac{1}{q}\right)^r q^{\epsilon r}.
\end{aligned}$$

We want to prove that this bound is at least  $q^k$ . Combining  $\epsilon \leq 1/2$  and  $q \geq (2/\epsilon)^{2/\epsilon}$ , we get  $q \geq 4^4 = 256$ . Substituting in the last equation,

$$\mu_m(r) \geq \left(\frac{\epsilon q^\epsilon}{5/4} \left(1 - \frac{1}{256}\right) \left(1 - \frac{1}{16}\right)\right)^r \geq \left(\frac{\epsilon}{2} q^\epsilon\right)^{2k/\epsilon} \geq \left(\left(\frac{\epsilon}{2}\right)^{2/\epsilon} \cdot q^2\right)^k \geq q^k.$$

It follows that there exists a vector  $\mathbf{v}$  such that  $|\mathcal{B}(\mathbf{v}, r) \cap \mathcal{A}| \geq q^k$ . This concludes the analysis for large square  $q$ .

**Case 2** (arbitrary  $q$ ): In this case, let  $c$  be the smallest even integer such that  $q' = q^c \geq (2/\epsilon)^{2/\epsilon}$ . Given  $k$ , let  $\mathcal{A}'[l', m', d']_{q'}$  be a code and  $r'$  a radius as given in Case 1 with rate  $m'/l' \geq \frac{3}{4}$ , relative distance  $d'/l' \geq 1/(2(\sqrt{q'}-1))$ ,  $r'/d' < \rho$  and  $\mu_{m'}(r') \geq (q')^k$ . We concatenate  $\mathcal{A}'$  with equidistant Hadamard code  $\mathcal{H}[q^c-1, c, q^c - q^{c-1}]_q$  to get a code  $\mathcal{A}[l, m, d]_q$  of rate

$$\frac{m}{l} = \frac{m'c}{l'(q^c-1)} \geq \frac{3c}{4q^c} > 0,$$

and relative distance

$$\frac{d}{l} = \frac{d'(q^c - q^{c-1})}{l'(q^c - 1)} \geq \left(1 - \frac{1}{q}\right) \cdot \frac{2(\sqrt{q^c} - 1)}{>} 0.$$

Now given a vector  $\mathbf{x}' \in \mathbb{F}_{q'}^{l'}$  that has  $(q')^k$  vectors in the ball of radius  $r'$  around it, the vector  $\mathbf{x} = \mathbf{x}' \diamond \mathcal{H}$  also has  $(q')^k$  vectors of  $\mathcal{A} \diamond \mathcal{H}$  in the ball of radius  $r = (q^c - q^{c-1})r'$  around it. Notice

that the ratio  $r/d = r'/d'$  does not change in this step, and therefore,  $r < \rho d$  holds true. This gives the code as claimed in the proposition.  $\square$

The following lemma stresses the algorithmic components of the code construction above and in particular stresses that we know how to construct the generator matrix of the  $k$ th code in the sequence, and how to sample dense balls from this code in polynomial time. (We also added one more parameter  $n$  to the algorithm to ensure that the block length of code  $\mathcal{A}$  is at least  $n$ . This additional parameter will be used in the proofs of Section 7, and is being added for purely technical reasons related to those proofs.)

**Proposition 25** *For every prime power  $q$  and every  $\rho > 2/3$ , there exist  $R > 0$ ,  $\delta > 0$  and a probabilistic algorithm that on input three integers  $k, n$  and  $s$ , outputs (in time polynomial in  $k, n, s$ ) three other integers  $l, m, r$ , a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  of a code  $\mathcal{A}$  and a center  $\mathbf{v} \in \mathbb{F}_q^l$  (of weight  $r$ ) such that*

1.  $m/l \geq R$ .
2.  $d(\mathcal{A})/l \geq \delta$ .
3.  $r < \rho \cdot d(\mathcal{A})$ .
4. with probability at least  $1 - q^{-s}$ , the ball  $\mathcal{B}(\mathbf{v}, r)$  contains at least  $q^k$  codewords of  $\mathcal{A}$ .
5.  $l \geq n$ .

**Proof:** Given inputs  $k, n, s$  to the algorithm, we consider the  $(\rho, k')$ -dense code  $\mathcal{A}_{k'}$  from Proposition 24, where  $k' = \max(n, k + s)$ . This code is the concatenation of an algebraic-geometry code  $\mathcal{A}'[l', m', d']_{q^c}$  and a Hadamard code  $\mathcal{H}[q^c - 1, c, q^c(1 - 1/q)]_q$ . Therefore a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$  for  $\mathcal{A}_{k'}$  can be constructed in time polynomial in  $k, n, s$ . Let  $r' = \lfloor \rho d' \rfloor$ ,  $r = r' \cdot q^c \cdot (1 - 1/q)$ , choose  $\mathbf{v}' \in \mathbb{F}_{q^c}^{l'}$  uniformly at random from  $\mathcal{B}(\mathbf{0}, r')$  and let  $\mathbf{v} = \mathbf{v}' \diamond \mathcal{H}$ . From the proof of Proposition 24 we know that the expected number of codewords in a randomly chosen ball  $\mathcal{B}(\mathbf{v}', r')$  is at least  $q^{k+s}$ . We now apply the Lemma 13 instantiating it with groups  $F = (\mathbb{F}_{q^c}^{l'}, +)$ ,  $G = (\mathcal{G}, +)$ , and set  $B = \mathcal{B}(\mathbf{0}, r')$ . Notice that for any center  $z = \mathbf{z} \in \mathbb{F}_{q^c}^{l'}$ , the set  $Bz$  is just the ball  $\mathcal{B}(\mathbf{z}, r')$  of radius  $r'$  centered in  $\mathbf{z}$ . From the proof of Proposition 24, the average size of  $G \cap Bz$  (i.e., the expected number of codewords in a random ball) is at least  $q^{k+s}$ . Setting  $\delta = q^{-s}$  in Lemma 13, and noting that  $B^{-1} = B = \mathcal{B}(\mathbf{0}, r')$ , we immediately get that  $\mathcal{B}(\mathbf{v}, r')$  contains at least  $q^k$  codewords with probability  $1 - q^{-s}$ . Finally, by Lemma 10, we get that the corresponding ball  $\mathcal{B}(\mathbf{v}, r)$  in  $\mathbb{F}_q^l$  contains at least  $q^k$  codewords from  $\mathcal{A}$ .  $\square$

We conclude with the following lemma analogous to Lemma 15 of Section 3.

**Lemma 26** *For every  $\rho > 2/3$  and prime power  $q$ , there exist  $R > 0$ ,  $\delta > 0$  and a probabilistic polynomial time algorithm that on input  $k, n, s$  outputs, (in time polynomial in  $k, n, s$ ), integers  $l, m, r$ , and matrices  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$ ,  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$  and a vector  $\mathbf{v} \in \mathbb{F}_q^l$  such that:*

1.  $m/l \geq R$ .
2.  $d(\mathcal{A})/l \geq \delta$ .
3.  $r < \rho \cdot d(\mathcal{A})$ .

4. With probability at least  $1 - q^{-s}$ ,  $\mathbf{T}(\mathcal{B}(\mathbf{v}, r) \cap \mathcal{A}) = \mathbb{F}_q^k$ , i.e., for every  $\mathbf{x} \in \mathbb{F}_q^k$  there exists a  $\mathbf{y} \in \mathcal{A}$  such that  $d(\mathbf{v}, \mathbf{y}) \leq r$  and  $\mathbf{y}\mathbf{T} = \mathbf{x}$ .
5.  $l \geq n$ .

The proof is identical to that of Lemma 15, with the use of Proposition 12 replaced by Proposition 25.

## 7 Hardness results for asymptotically good codes

In this section we prove hardness results for the decoding problem even when restricted to asymptotically good codes, and the hardness of approximating the minimum distance problem with linear (in the block length) additive error. First, we define a restricted version of GAPRNC where the code is required to be asymptotically good.

**Definition 27** For  $R, \delta \in (0, 1)$ , let  $(R, \delta)$ -restricted  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  be the restriction of  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  to instances  $(\mathbf{A} \in \mathbb{F}_q^{k \times n}, \mathbf{v}, t)$  satisfying  $k \geq R \cdot n$  and  $t \geq \delta \cdot n$ .

We call  $\delta$  relative error weight. Notice that for every constant  $\rho$ , the minimum distance of the code satisfies  $d > t/\rho \geq \delta n/\rho$ , therefore the code is asymptotically good with information rate at least  $k/n \geq R > 0$  and relative distance at least  $d/n > \delta/\rho > 0$ . In particular, when  $\rho \leq 1$ , the relative distance is strictly bigger than  $\delta$ .

In the following subsections, we prove that  $(R, \delta)$ -restricted GAPRNC is NP-hard (under RUR reductions). Then we use this result to prove that the minimum distance of a linear code is hard to approximate additively, even to within a *linear additive error* relative to the block length of the code. The proofs will be essentially the same as those for non-asymptotically good codes. The main differences are the following:

- We use the asymptotically good dense code construction from Section 6. Before we relied on dense codes that were not asymptotically good.
- We reduce from a restricted version of the nearest codeword problem in which the error weight is required to be a constant fraction of the block length.

In the following subsection we observe that a result of Håstad [13] gives us hardness of NCP with this additional property. Then we use this result to prove the hardness of the  $(R, \delta)$ -restricted GAPRNC problem, and the hardness of GAPDISTADD.

### 7.1 The restricted nearest codeword problem

We define a restricted version of the nearest codeword problem in which the target distance is required to be at least a linear fraction of the block length.

**Definition 28** For  $\tau \in (0, 1)$ , let  $\tau$ -restricted  $\text{GAPNCP}_{\gamma, q}$  ( $\tau$ -GAPNCP $_{\gamma, q}$  for short) be the restriction of  $\text{GAPNCP}_{\gamma, q}$  to instances  $(\mathbf{A} \in \mathbb{F}_q^{k \times n}, \mathbf{v}, t)$  satisfying  $t \geq \tau \cdot n$ .

We will need the fact that  $\tau$ -GAPNCP $_{\gamma, q}$  is NP-hard. This result does not appear to follow from the proof of [3]. Instead we observe that this result easily follows from a recent and extremely powerful result of Håstad [13].

**Theorem 29 ([13])** *For every Abelian group  $G$  and for every  $\tau \in (0, (1-1/|G|)/2)$ , given a system of  $n$  linear equations over  $G$ , it is NP-hard to distinguish instances in which  $(1-\tau)n$  equations can be satisfied, from those in which at most  $\left(\frac{1}{|G|} + \tau\right)n$  equations can be satisfied<sup>13</sup>.*

Notice that a system of  $n$  linear equations where  $s$  is the maximum number of simultaneously satisfiable equations corresponds to an instance of the nearest codeword problem where the distance of the target from the code is  $n-s$ . So, applied to group  $G = (\mathbb{F}_q, +)$ , and phrased in coding theoretic terms the theorem says that for every prime power  $q$ , it is hard to tell whether the distance of the target from the code is at most  $\tau n$  or more than  $(1-1/q-\tau)n$ . In other words,  $\tau$ -GAPNCP $_{\gamma,q}$  is NP-hard for every constant approximation factor  $\gamma \leq \frac{1-(1/q)-\tau}{\tau}$ . In particular,  $\tau$ -GapNCP $_{\gamma,q}$  is NP-hard for  $\gamma = \frac{1}{2\tau} - 1$ . This gives the following version of Theorem 29.

**Corollary 30** *For every  $\gamma > 1$ , there exists a  $\tau > 0$  such that  $\tau$ -GAPNCP $_{\gamma,q}$  is NP-hard for all prime powers  $q$ .*

Notice that the  $\tau$  in the corollary is independent of the alphabet size.

## 7.2 Inapproximability of the restricted relatively near codeword problem

In this subsection we prove the hardness of the  $(R, \delta)$ -restricted GAPRNC $_{\gamma,q}^{(\rho)}$  problem. The proof is the same as that of Theorem 16 with the following modifications: (1) instead of using Lemma 15 (i.e., the dense code family based on asymptotically bad MDS codes) we use the asymptotically good dense code family from Lemma 26; (2) we consider only  $\rho > 2/3$ , so that Lemma 26 holds true; (3) we use Corollary 30 to select a  $\tau > 0$  such that  $\tau$ -GAPNCP $_{\gamma,q}$  is NP-hard; (4) we assume that the GAPNCP $_{\gamma,q}$  instance  $(\mathbf{C}, \mathbf{v}, t)$  we are reducing is in fact a  $\tau$ -GAPNCP $_{\gamma,q}$  instance. Details follow.

**Theorem 31** *For every prime power  $q$ , and constants  $\rho' > 2/3$  and  $\gamma' \geq 1$ , there exists  $R' > 0$  and  $\delta' > 0$  such that  $(R', \delta')$ -restricted GAPRNC $_{\gamma',q}^{(\rho')}$  is hard for NP under RUR-reductions with error probability exponentially small in a security parameter.*

**Proof:** Fix a prime power  $q$  and real  $\rho > 2/3$ , and let  $R > 0$  and  $\delta > 0$  be the corresponding constants such that Lemma 26 holds true. Let  $\epsilon > 0$  be an arbitrarily small constant, and  $\gamma' \geq 1$  the approximation factor for which we want to prove the NP-hardness of the restricted GAPRNC $_{\gamma',q}$  problem. Let  $\gamma = (2 + 1/\epsilon)\gamma'$ , and let  $\tau > 0$  be the constant from Corollary 30 such that  $\tau$ -GAPNCP $_{\gamma,q}$  is NP-hard, e.g.,  $\tau = 1/(2\gamma + 2)$ . We prove that  $(R', \delta')$ -restricted GAPRNC $_{\gamma',q}^{(\rho')}$  is NP-hard for  $\rho' = \rho(1 + \epsilon)$ ,  $\delta' = \min(\delta/2, \tau)$  and  $R' = R/(1 + \frac{2}{\epsilon\delta} + \frac{2}{\tau} + \frac{1}{\epsilon})$ . Since  $\rho$  can be arbitrarily close to  $2/3$ , and  $\epsilon$  arbitrarily close to 0, this proves the NP-hardness of  $(R', \delta')$ -restricted GAPRNC $_{\gamma',q}^{(\rho')}$  for any  $\rho' > 2/3$ .

We prove the NP-hardness of  $(R', \delta')$ -restricted GAPRNC $_{\gamma',q}^{(\rho')}$  by reduction from  $\tau$ -GAPNCP $_{\gamma,q}$ . Let  $(\mathbf{C}, \mathbf{u}, t)$  be an instance of  $\tau$ -GAPNCP $_{\gamma,q}$ , where  $\mathbf{C} \in \mathbb{F}_q^{k \times n}$ . We invoke the algorithm of Lemma 26 on input  $k, n$  and  $s$ , to find integers  $l, m, r$ , a generator matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times l}$ , a mapping matrix  $\mathbf{T} \in \mathbb{F}_q^{l \times k}$ , and a vector  $\mathbf{v} \in \mathbb{F}_q^l$  such that

<sup>13</sup>Håstad's result has the further property that every linear equation only involves three variables, but we don't need this extra property.

1.  $r < \rho \cdot d(\mathcal{A})$ ,
2. for every  $\mathbf{x} \in \mathbb{F}_q^k$  there exists a  $\mathbf{y} \in \mathcal{A}$  satisfying  $d(\mathbf{v}, \mathbf{y}) \leq r$  and  $\mathbf{y}\mathbf{T} = \mathbf{x}$ ,
3.  $m \geq Rl$ ,
4.  $d(\mathcal{A}) \geq \delta l$ ,
5.  $l \geq n$ , and therefore  $l \geq t$  too.

where the second condition holds with probability  $1 - q^{-s}$ . Notice that, since any sphere containing more than a single codeword must have radius at least  $d(\mathcal{A})/2$ , the relative radius must be at least

$$\frac{r}{l} \geq \frac{d(\mathcal{A})}{2l} \geq \frac{\delta}{2}.$$

The reduction proceeds as in the proof of Theorem 16. Let  $b = \lceil \frac{r}{t} \rceil$ ,  $a = \lceil \frac{bt}{\epsilon r} \rceil$ ,  $t' = ar + bt$ , and define code  $\mathbf{C}'$  and target  $\mathbf{u}'$  as in (10) and (11). The output of the reduction is  $(\mathbf{C}', \mathbf{u}', t')$ . We want to prove that the map  $(\mathbf{C}, \mathbf{u}, t) \mapsto (\mathbf{C}', \mathbf{u}', t')$  is a valid (RUR) reduction from  $\tau$ -GAPNCP $_{\gamma, q}$  to  $(R', \delta')$ -restricted GAPRNC $_{\gamma', q}^{(\rho')}$ . The proof of the following facts is exactly the same as in Theorem 16 (Equations (12), (13) and (14)):

- $t' < \rho' \cdot d(\mathcal{C}')$
- if  $d(\mathbf{u}, \mathcal{C}) \leq t$ , then  $d(\mathbf{u}', \mathcal{C}') \leq t'$
- if  $d(\mathbf{u}, \mathcal{C}) > \gamma \cdot t$ , then  $d(\mathbf{u}', \mathcal{C}') > \gamma' \cdot t'$

It remains to prove that GAPRNC instance  $(\mathbf{C}', \mathbf{u}', t')$  satisfies the  $(R', \delta')$  restriction. Notice that the code  $\mathbf{C}'$  has block length  $n' = al + bn$ . Therefore the relative error weight  $t'/n'$  satisfies

$$\frac{t'}{n'} = \frac{ar + bt}{al + bn} \geq \min\left(\frac{r}{l}, \frac{t}{n}\right) \geq \min(\delta/2, \tau) = \delta'.$$

Finally, the information content of  $\mathcal{C}'$  is  $m$ , i.e., the same as code  $\mathcal{A}$ . Therefore, the rate is  $m/n' \geq R(l/n')$  and in order to show that the code is asymptotically good we need to prove that  $n' = O(l)$ . In fact,

$$a = \left\lceil \frac{bt}{\epsilon r} \right\rceil \leq 1 + \frac{\lceil \frac{r}{t} \rceil t}{\epsilon r} \leq 1 + \frac{t+r}{\epsilon r} \leq 1 + \frac{1}{\epsilon} + \frac{l}{\epsilon r} \leq 1 + \frac{1}{\epsilon} + \frac{2}{\epsilon \delta}$$

$$b = \left\lceil \frac{r}{t} \right\rceil \leq \frac{t+r}{t} \leq \frac{2l}{\tau n}$$

and

$$n' = al + bn \leq \left(1 + \frac{1}{\epsilon} + \frac{2}{\epsilon \delta} + \frac{2}{\tau}\right) l.$$

This proves that the rate of the code is at least

$$\frac{m}{n'} \geq \frac{R}{1 + \frac{1}{\epsilon} + \frac{2}{\epsilon \delta} + \frac{2}{\tau}} = R'.$$

□

### 7.3 Minimum distance with linear additive error

In Lemma 19 we proved that the minimum distance problem is hard to approximate within some constant factor. The proof was by reduction from GAPRNC. The same reduction, when applied to the restricted GAPRNC problem gives an inapproximability result for the minimum distance problem with linear additive error. In particular, Lemma 19 reduces  $(R, \delta)$ -restricted  $\text{GAPRNC}_{1/\rho, q}^{(\rho)}$  to  $\text{GAPDISTADD}_{\tau, q}$  with  $\tau = \delta \cdot ((1/\rho) - 1) > \delta \cdot (1 - \rho)$ . This result is formally proved in the following theorem.

**Theorem 32** *For every prime power  $q$ , there exists a  $\tau > 0$  such that  $\text{GAPDISTADD}_{\tau, q}$  is hard for NP under polynomial RUR-reductions with soundness error exponentially small in a security parameter.*

**Proof:** The proof is by reduction from  $(R, \delta)$ -restricted  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  with  $\rho < 1$ ,  $\gamma = 1/\rho$ , and  $\tau = \delta(1 - \rho)$ . The reduction is the same as in the proof of Lemma 19. On input  $(\mathbf{C}, \mathbf{u}, t)$ , the reduction outputs  $(\mathbf{C}', t)$  where  $\mathbf{C}'$  is the code defined in (15). We know from the proof of Lemma 19 that  $(\mathbf{C}, \mathbf{u}, t) \mapsto (\mathbf{C}', t)$  is a valid reduction from  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  to  $\text{GAPDIST}_{\gamma, q}$ . We show that if  $(\mathbf{C}', t)$  is a YES (resp. NO) instance of  $\text{GAPDIST}_{\gamma, q}$ , then it is also a YES (resp. NO) instance of  $\text{GAPDISTADD}_{\tau, q}$ . This proves that  $(\mathbf{C}, \mathbf{u}, t) \mapsto (\mathbf{C}', t)$  is also a valid reduction from  $\text{GAPRNC}_{\gamma, q}^{(\rho)}$  to  $\text{GAPDISTADD}_{\tau, q}$ .

The case of YES instances is trivial, because the definition of YES instances in  $\text{GAPDIST}_{\gamma, q}$  and  $\text{GAPDISTADD}_{\tau, q}$  is the same, namely  $d(\mathbf{C}') \leq t$ . So, assume  $(\mathbf{C}', t)$  is a NO instance of  $\text{GAPDIST}_{\gamma, q}$ . Then,

$$d(\mathbf{C}') > \gamma \cdot t = t + \left(\frac{1}{\rho} - 1\right)t \geq t + \left(\frac{1}{\rho} - 1\right)\delta n \geq t + \tau n,$$

and  $(\mathbf{C}', t)$  is a NO instance of  $\text{GAPDISTADD}_{\tau, q}$ .  $\square$

## 8 Other reductions

We proved that approximating the minimum distance problem is hard for NP under RUR-reductions, i.e. probabilistic reductions that map NO instances to NO instances, and YES instances to YES instances with high probability. (The same is true for all other reductions presented in this paper, as well as the hardness proof for the shortest vector problem in [2, 18].)

An obvious question is whether it is possible to remove the randomization and make the reduction deterministic. (Observe that a deterministic NP-hardness result is known for the exact version of the minimum distance problem [23, 22]. In contrast, for the shortest vector problem, even the exact version is known to be NP-hard only under randomized reductions [2].) We notice that our reductions (as well as the Ajtai-Micciancio ones for SVP) use randomness in a very restricted way. Namely, the only part of the reduction where randomness is used is the proof of Lemma 15 (and Lemma 26 for asymptotically good codes). The construction in the lemma depends only on the input size, and not on the particular input instance we are reducing. So, if the construction succeeds, the reduction will faithfully map all YES instances (of the appropriate size) to YES instances. Therefore, the statements in Lemma 15 and Lemma 26 can be easily modified to obtain hardness results for NP under *deterministic non-uniform* reductions, i.e. reductions that take a polynomially

sized advice that depends only on the input size<sup>14</sup>. (The advice is given by the sphere center  $\mathbf{v}$  and transformation matrix  $\mathbf{T}$  satisfying Lemma 15 and Lemma 26.)

**Corollary 33** *For every finite field  $\mathbb{F}_q$  the following holds:*

- *For every  $\rho > 1/2$ , and  $\gamma \geq 1$   $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  is hard for NP under nonuniform polynomial reductions. Therefore, no nonuniform polynomial time (P/poly) algorithm exists for  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  unless  $\text{NP} \subseteq \text{P/poly}$ .*
- *For every  $\gamma > 1$ ,  $\text{GAPDIST}_{\gamma,q}$  is hard for NP under nonuniform polynomial reductions. Therefore, no P/poly algorithm exists for  $\text{GAPDIST}_{\gamma,q}$  unless  $\text{NP} \subseteq \text{P/poly}$ .*
- *For every  $\epsilon > 0$  and  $\gamma(n) = 2^{\log^{(1-\epsilon)} n}$ ,  $\text{GAPDIST}_{\gamma,q}$  is hard for NP under nonuniform quasi-polynomial reductions. Therefore, no nonuniform quasi-polynomial time (QP/poly) algorithm exists for  $\text{GAPDIST}_{\gamma,q}$  unless  $\text{NP} \subseteq \text{QP/poly}$ .*
- *For every  $\rho > 1/2$ , and  $\gamma \geq 1$  there exists  $R > 0$  and  $\delta > 0$  such that  $(R, \delta)$ -restricted  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  is hard for NP under nonuniform polynomial reductions. Therefore, no P/poly algorithm exists for  $(R, \delta)$ -restricted  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  unless  $\text{NP} \subseteq \text{P/poly}$ .*
- *There exists a  $\tau > 0$  such that  $\text{GAPDISTADD}_{\tau,q}$  is hard for NP under nonuniform polynomial reductions. Therefore, no P/poly algorithm exists for  $\text{GAPDISTADD}_{\tau,q}$  unless  $\text{NP} \subseteq \text{P/poly}$ .*

We notice also that a *uniform* deterministic construction satisfying the properties of Lemma 15 and Lemma 26 would immediately give a proper NP-hardness result (i.e. hardness under deterministic Karp reductions) for all problems considered in this paper. Interestingly, for the shortest vector problem, Micciancio [18] (see also [19]) showed that a deterministic NP-hardness result is possible under a reasonable (but unproven) number theoretic conjecture regarding the distribution of smooth numbers (i.e., numbers with no large prime factors). No deterministic proof under similar number theoretic conjectures is known for the coding problems studied in this paper.

Finally, we notice that all our results rely on the fact that the code is given as part of the input. Thus it is conceivable that for every error-correcting code, there exists a fast algorithm to correct errors (say up to the distance of the code), however, this algorithm may be hard to find (given a description of the code). A result along the lines of the one of Bruck and Naor [7], showing that there are specific sequences of codes for which nearest codewords are hard to find (even if the code can be arbitrarily preprocessed) would be desirable to fix this gap in our knowledge. Feige and Micciancio [10] recently showed that the nearest codeword problem with preprocessing is hard to approximate within any factor less than  $5/3$ . Inapproximability results for NCP with preprocessing are interesting because they can be combined with our reduction (Theorem 16) to prove the inapproximability of RNC with preprocessing. Specifically, Theorem 16 gives a reduction from  $\text{GAPNCP}_{\gamma,q}$  to  $\text{GAPRNC}_{\gamma',q}^{(\rho)}$  with  $\rho = (1 + \epsilon)/2$  and  $\gamma' < \gamma/(2 + 1/\epsilon)$ , with the property that the GAPRNC code depends only on the GAPNCP code (and not the target vector). So, if  $\text{GAPNCP}_{\gamma,q}$  is hard to solve even when the code can be preprocessed, then also  $\text{GAPRNC}_{\gamma',q}^{(\rho)}$

---

<sup>14</sup>Since our reduction achieves exponentially small error probability, hardness under non-uniform reductions also follows from general results about derandomization [1]. However, the ad-hoc derandomization method we just described is more efficient and intuitive.

with preprocessing cannot be efficiently solved. Unfortunately, in order to get non trivial hardness results for  $\text{GAPRNC}_{\gamma',q}^{(\rho)}$  with preprocessing (i.e., results with  $\gamma' \geq 1$ ) using this method, one needs to start from inapproximability results for  $\text{GAPNCP}_{\gamma,q}$  with preprocessing with  $\gamma > 2$ . So, inapproximability factors  $\gamma < 5/3$  proved in [10] do not suffice. Very recently, the results of [10] have been improved by Regev [20] to  $\gamma < 3$ , giving the first inapproximability results for  $\text{RNC}^{(\rho)}$  with preprocessing. Notice that applying the reduction from Theorem 16 to  $\text{GAPNCP}_{\gamma,q}$  with  $\gamma < 3$  allows to establish the inapproximability (within some constant factor  $\gamma' < 3/(2 + 1/\epsilon)$ ) of  $\text{GAPRNC}^{(\rho)}$  for  $\rho = (1 + \epsilon)/2$ . Notice that in order to get  $\gamma' \geq 1$ , one needs  $\epsilon > 1$ , and therefore  $\rho > 1$ . Interestingly, [20] also shows how to modify the reduction in Theorem 16 to get inapproximability of  $\text{RNC}^{(\rho)}$  with preprocessing for any  $\rho > 1/2$ . The improvement is based on the following two observations. (The reader is referred to [20] for details.)

1. The lower bound  $d(\mathcal{C}', \mathbf{u}') \geq b \cdot d(\mathcal{C}, \mathbf{u})$  in (14) can be strengthened to

$$d(\mathcal{C}', \mathbf{u}') \geq a \cdot d(\mathcal{A}, \mathbf{v}) + b \cdot d(\mathcal{C}, \mathbf{u}).$$

2. The center of the sphere  $\mathbf{v}$  is at distance at least

$$d(\mathcal{A}, \mathbf{v}) \geq \min(\|\mathbf{v}\|, d(\mathcal{A}) - \|\mathbf{v}\|) \geq \left(\frac{1}{\rho} - 1\right) r$$

from the code  $\mathcal{A}$ .

Based on these two observations, Regev shows that (with a different choice of parameters  $a$  and  $b$ ) the reduction given in the proof of Theorem 16 maps  $\text{GAPNCP}_{\gamma,q}$  to  $\text{GAPRNC}_{\gamma',q}^{(\rho)}$  for any  $\gamma' < \left(1 - \frac{1}{2\rho}\right) \gamma + \frac{1}{2\rho}$ . Combined with the improved inapproximability results for  $\text{GAPNCP}$  with preprocessing, this shows that  $\text{GAPRNC}_{\gamma,q}^{(\rho)}$  with preprocessing is NP-hard (under RUR-reductions as usual) for any  $\rho > 1/2$  and  $\gamma < 3 - 1/\rho$ .

We conclude with a brief discussion of the main problem left open in this work. Our hardness results for  $\text{RNC}^{(\rho)}$  hold only for  $\rho > 1/2$ . For the case of general (not asymptotically good) codes, we showed that one can make  $\rho$  arbitrarily close to  $1/2$ . However, we cannot achieve equality  $\rho = 1/2$ .  $\text{RNC}^{(1/2)}$  (also known as the “bounded distance decoding” problem, BDD) is arguably the most relevant decoding problem in practice, and the one for which most known polynomial time algorithms (for specific families of codes, e.g., RS codes) work. The problems  $\text{RNC}^{(1/2)}$  and  $\text{RNC}^{(\rho)}$  for  $\rho > 1/2$  are qualitatively different, because for  $\rho \leq 1/2$  the solution to the decoding problem is guaranteed to be unique. Our reduction relies on the construction of a code  $\mathcal{A}$  and a sphere  $\mathcal{B}(\mathbf{v}, r)$  of radius  $r < \rho \cdot d(\mathcal{A})$  such that  $\mathcal{B}(\mathbf{v}, r)$  contains several (in fact, exponentially many) codewords. Clearly, for  $\rho \leq \frac{1}{2}$ , no sphere of radius  $r < \rho \cdot d(\mathcal{A})$  can contain more than a single codeword. So, the bound  $\rho > 1/2$  seems an intrinsic limitation of our reduction technique.<sup>15</sup> We leave determining the computational complexity of BDD as an open problem.

## References

- [1] L. M. Adleman. Two theorems on random polynomial time. In *Proc. 19th Symposium on Foundations of Computer Science*, pages 75–83, 1978.

---

<sup>15</sup>The same problem on integer lattices is also the current bottleneck to extend the inapproximability factor of the shortest vector problem beyond  $\sqrt{2}$  [18].

- [2] M. Ajtai. The shortest vector problem in  $L_2$  is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 10–19, Dallas, Texas, May 23–26 1998.
- [3] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, Apr. 1997. Preliminary version in FOCS’93.
- [4] S. Arora and C. Lund. *Approximation algorithms for NP-hard problems*, chapter 10, Hardness of Approximation. PWS Publishing, Boston, 1996.
- [5] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
- [6] P. Berman and M. Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proceedings of the 13th annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’02*, pages 74–83, San Francisco, California, 2002.
- [7] J. Bruck and M. Naor. The hardness of decoding linear codes with preprocessing. *IEEE Transactions on Information Theory*, 36(2):381–385, Mar. 1990.
- [8] I. Dinur, G. Kindler, R. Raz, and S. Safra. An improved lower bound for approximating CVP. Manuscript, 1999.
- [9] I. Dinur, G. Kindler, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *39th annual symposium on foundations of computer science*, Palo Alto, California, Nov. 7–10 1998. IEEE.
- [10] U. Feige and D. Micciancio. The inapproximability of lattice and coding problems with preprocessing. In *IEEE Conference on Computational Complexity - CCC 2002*, pages 44–52, Montreal, Canada, May 21–23 2002. IEEE.
- [11] G. D. Forney Jr. *Concatenated codes*. Number 37 in Research Monograph. MIT Press, Cambridge, Massachusetts, 1966.
- [12] J. Håstad. Clique is hard to approximate within  $n^{1-\epsilon}$ . *Acta Mathematica*, 182:105–142, 1999.
- [13] J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001. Preliminary version in Proc. of STOC’97.
- [14] K. Jain, M. Sudan, and V. Vazirani. Personal communication, May 1998.
- [15] D. S. Johnson. *Handbook of Theoretical Computer Science*, volume A (Algorithms and Complexity), chapter 2, A catalog of complexity classes, pages 67–161. Elsevier, 1990.
- [16] J. Justesen and T. Høholdt. Bounds on list decoding of MDS codes. *IEEE Transactions on Information Theory*, 47(4):1604–1609, May 2001.
- [17] F. MacWilliams and N. Sloane. *The theory of error-correcting codes*. North-Holland, Amsterdam, The Netherlands, 1981.

- [18] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, Mar. 2001. Preliminary version in FOCS 1998.
- [19] 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.
- [20] O. Regev. Improved inapproximability of lattice and coding problems with preprocessing. Manuscript, Aug. 2002.
- [21] M. Tsfasman and S. Vlăduț. *Algebraic-Geometric Codes*. Kluwer, Dordrecht, 1991.
- [22] A. Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *Proceedings of the twenty-ninth annual ACM Symposium on Theory of computing, STOC'97*, pages 92–109, El Paso, Texas, May 4–6 1997. ACM.
- [23] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, Nov. 1997.
- [24] V. Zinoviev and S. Litsyn. On codes exceeding the gilbert bound. *Problems of Information Transmission*, 21(1):109–111, 1985. (in Russian).