Hardness of Approximating the Minimum Distance of a Linear Code

Ilya Dumer* Daniele Micciancio† Madhu Sudan‡

Abstract

*We show that the minimum distance of a linear code (or equivalently, the weight of the lightest codeword) is not approximable to within any constant factor in random polynomial time (*RP*), unless* NP *equals* RP*. 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 $\epsilon > 0$, where n denotes the block length of the code. Our results hold for codes over every finite field, including the special case of binary codes. In the process we show that the nearest codeword problem is hard to solve even under the promise that the number of errors is (a constant factor) smaller than the distance of the code. This is a particularly meaningful version of the nearest codeword problem.*

Our results strengthen (though using stronger assumptions) a previous result of Vardy who showed that the minimum distance is NP*-hard to compute exactly. 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.

*College of Engineering, University of California at Riverside. Riverside, CA 92521, USA. Email: dumer@ee.ucr.edu. Research supported in part by NSF grant NCR-9703844.

†Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. 545 Technology Square, Cambridge, MA 02139, USA. Email: miccianc@theory.lcs.mit.edu. Research supported in part by DARPA grant DABT63-96-C-0018.

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

An error-correcting code $\mathcal{C}$ over a q-ary alphabet Σ of block length n, is a collection of strings from Σ^n. The Hamming distance between two strings $\mathbf{x}, \mathbf{y} \in \Sigma^n$ is the number $\Delta(\mathbf{x}, \mathbf{y})$ of coordinates in which $\mathbf{x}$ and $\mathbf{y}$ differ. The (Hamming) weight of a string $\mathbf{x}$ is $\mathrm{wt}(\mathbf{x}) = \Delta(\mathbf{x}, \mathbf{0})$. The (minimum) distance of the code, denoted $\Delta(\mathcal{C})$, is the minimum over all pairs of distinct strings $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ of the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$. The information content of the code is the quantity $\log_q |\mathcal{C}|$, which counts the number of message symbols that can be encoded by an element of $\mathcal{C}$. If q is a prime power, and $\mathbb{F}_q$ denotes the finite field on q elements, then by setting $\Sigma = \mathbb{F}_q$ it is possible to think of $\Sigma^n = \mathbb{F}_q^n$ as a vector space. A code over $\mathbb{F}_q$ is *linear* if it is a linear subspace of $\Sigma^n = \mathbb{F}_q^n$. For such a code, the information content is just its dimension as a vector space and the minimum distance equals the weight of the lightest non-zero codeword. It is customary to refer to a linear code of block length n, dimension k and minimum distance d as an $[n, k, d]_q$ code. We use an $n \times k$ matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$ of rank k to define a linear code $\mathcal{C}_{\mathbf{A}} = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{F}_q^k\}$ of length n and dimension k.

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 matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$ find the minimum distance of the code $\mathcal{C}_{\mathbf{A}} = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{F}_q^k\}$. It is easy to see that a code with minimum distance d can unambiguously correct any error vector of weight $\lfloor \frac{d-1}{2} \rfloor$ or less. (For details on the computational complexity of the error correction problem see the next paragraph.) Therefore, computing the minimum distance of a code is obviously related to the problem of evaluating its error correction capability. The central nature of this parameter makes this 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 [1] known for any $q < 46$

[1] For squares $q \geq 49$, linear AG codes can perform better than random ones [14] and are constructed in polynomial time. For any $q \geq 46$ it is still

(in particular, for $q = 2$). Such a choice is expected to produce a code of large distance, but no efficient methods are known to lower bound the distance of a code produced in this manner. A polynomial time algorithm to compute the distance would be the ideal solution to this problem, as it could be used to construct good error correcting codes by choosing a matrix at random and checking if the associated code has a large minimum distance. 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 [7] in 1978 who conjectured it to be NP-complete. This conjecture was finally resolved in the affirmative by Vardy ([15]) in 1997. ([15] also gives further motivations and detailed account of prior work on this problem.) We examine the approximability of this parameter and show that it is hard to approximate the minimum distance to within any constant factor, unless NP = RP (i.e., every problem in NP has a polynomial time probabilistic algorithm that always reject NO instances and accepts YES instances with high probability). Under the stronger assumption that NP does not have random quasi-polynomial time[2] algorithms (RQP), we get 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 factor 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 non-approximability of the shortest lattice vector problem (SVP) due to Micciancio [13] which in turn is based on Ajtai's proof of the hardness of SVP [3].

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. Informally, the error-correction problem addresses the computational complexity of recovering a codeword from a "received word" that is close to the codeword in Hamming distance. The simplest formulation of the error-correction problem is the Nearest Codeword Problem (NCP) (also known as the "maximum likelihood decoding problem"). Here, the input instance consists of a linear code given by its matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$ and a received word $\mathbf{x} \in \mathbb{F}_q^n$ and the goal is to find the nearest codeword $\mathbf{y} \in \mathcal{C}_{\mathbf{A}}$ to $\mathbf{x}$. The NCP is a well-studied problem: Berlekamp et al. [7] showed that it is NP-hard; and more recently Arora, Babai, Stern and Sweedyk [2] showed that the distance of the received word to the nearest codeword is hard (unless NP $\subseteq$ QP, deterministic quasi-polynomial time) to approximate to within a factor of $2^{\log^{(1-\epsilon)} n}$, for any $\epsilon > 0$.

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* for *arbitrary amounts of error*. In contrast, the positive results from coding theory show how to perform error-correction in *specific linear codes* for a *small amount of error relative to the distance of the code*. 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 [15], but only the former has seen some progress [6]. One problem that has been defined to study the latter phenomenon is the "Bounded distance decoding problem" (BDD, see [15]). This is a special case of the NCP where the error is guaranteed (or "promised") to be less than half the minimum distance of the code. 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.

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 ρ, RNC$^{(\rho)}$ is the following problem:

Given a generator matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$ of a linear code $\mathcal{C}_{\mathbf{A}}$ of 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, the algorithm is expected to work only when the amount of error that occurs is limited in proportion to the error that the code was designed to tolerate.)

Both the nearest codeword problem (NCP) and the bounded distance decoding problem (BDD) are special cases of RNC$^{(\rho)}$: NCP = RNC$^{(\infty)}$ while BDD = RNC$^{(\frac{1}{2})}$. Till recently, not much was known about RNC$^{(\rho)}$ for constants $\rho < \infty$, leave alone $\rho = \frac{1}{2}$ (i.e., the BDD problem). No finite upper bound on ρ can be easily derived from the Arora et al.'s NP-hardness proof for NCP [2]. (In other words, their proof does not seem to hold for RNC$^{(\rho)}$ for any $\rho < \infty$.) It turns out, as observed by Jain et al. [9], that Vardy's proof of the NP-hardness of the minimum distance problem also shows the NP-hardness of 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 (for random reductions) of RNC$^{(\rho)}$ for every $\rho > \frac{1}{2}$ bringing us much closer to an eventual (negative?) resolution of the bounded distance decoding problem.

possible to do better than random codes using an exponential procedure [16].

[2] $f(n)$ is quasi-polynomial in n if it grows slower than $2^{\log^c n}$ for some constant c.

1.3. Results and Techniques.

The main result of this paper (see Theorem 15) is that approximating the minimum distance problem within any constant factor is hard for NP under polynomial *reverse unfaithful random* reductions (RUR-reductions, [10]), and approximating it within $2^{\log^{(1-\epsilon)} n}$ is hard under quasi-polynomial RUR-reductions. These are probabilistic reductions that maps NO instances always to NO instances and YES instances to YES instances with high probability. The probability a YES instance is not mapped to a YES instance is called the *soundness error* and in all reductions presented in this paper it can be made exponentially small in a security parameter s in $\mathrm{poly}(s)$ time. Although not a proper NP-hardness result (i.e., hardness under deterministic polynomial reductions), hardness under polynomial RUR-reductions also gives evidence of the intractability of a problem as the existence of a (random) polynomial time algorithm to solve the hard problem would imply $\mathrm{NP} = \mathrm{RP}$ (random polynomial time), i.e. every problem in NP would have a probabilistic polynomial algorithm that always rejects NO instances and accepts YES instances with high probability. Similarly, hardness for NP under quasi-polynomial RUR-reductions implies that the hard problem cannot be solved in RQP unless $\mathrm{NP} \subseteq \mathrm{RQP}$ (random quasi-polynomial time).

In order to prove these results, we first study the "Relatively near Codeword Problem" and show that the optimization version of $\mathrm{RNC}^{(\rho)}$ is hard to approximate to within any constant factor γ for any $\rho > 1/2$ unless $\mathrm{NP} = \mathrm{RP}$ (see Theorem 9). In particular $\mathrm{RNC}^{(\rho)}$ is hard to approximate to within $\gamma = 1/\rho$. This problem immediately reduces to approximating the minimum distance of a code within $\gamma = 1/\rho$. This gives a first inapproximability result for the minimum distance problem within some constant factor $\gamma > 1$. We then use tensor product constructions to "amplify" the constant and prove the claimed hardness results for the minimum distance problem.

The hardness of approximating the relatively near codeword problem $\mathrm{RNC}^{(\rho)}$ for $\rho > 1/2$ is obtained by adapting a technique of Micciancio [13], which is in turn based on the work of Ajtai [3] (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 adaption is straightforward; in fact, some of the proofs are even easier in our case due to the difference. 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 $\mathrm{poly}(k)$ time, an integer d, a lattice $\mathcal{L}$ in $\mathbb{Z}^k$ with minimum distance d and a vector $\mathbf{v} \in \mathbb{Z}^k$ such that a (Euclidean) ball of radius $\rho \cdot d$ around $\mathbf{v}$ contains at least 2^{k^ϵ} vectors from $\mathcal{L}$ (where $\rho < 1$ and $\epsilon > 0$ are some constants independent of k).

In our case we are faced with a similar problem with $\mathbb{Z}^k$ replaced by $\mathbb{F}_q^k$ and Euclidean distance being 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 codes. (This construction has been used in the coding theory literature to demonstrate limitations to the "list-decodability" of Reed-Solomon codes [11].) We then translate the result to small alphabets using the well-known method of concatenating codes [8].

2. Notations and problem definition

For a vector $\mathbf{v} \in \mathbb{F}_q^n$ and set $S \subseteq \mathbb{F}_q^n$, let $\Delta(\mathbf{v}, S) = \min_{\mathbf{w} \in S}\{\Delta(\mathbf{v}, \mathbf{w})\}$ be the (Hamming) distance between $\mathbf{v}$ and S. For vector $\mathbf{v} \in \mathbb{F}_q^n$ and positive integer r, let $\mathcal{B}(\mathbf{v}, r) = \{\mathbf{w} \in \mathbb{F}_q^n | \Delta(\mathbf{v}, \mathbf{w}) \leq r\}$ be the ball of radius r centered in $\mathbf{v}$. Given a generator matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, we consider the linear code $\mathcal{C}_\mathbf{A} = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \in \mathbb{F}_q^k\}$ of distance $\Delta(\mathcal{C}_\mathbf{A}) = \min\{\mathrm{wt}(\mathbf{A}\mathbf{x}) \mid \mathbf{x} \neq 0\}$.

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 γ.

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

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

In other words, given a code $\mathbf{A}$ and an integer d with the promise that either $\Delta(\mathcal{C}_\mathbf{A}) \leq d$ or $\Delta(\mathcal{C}_\mathbf{A}) > \gamma \cdot d$, 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 one hand, if one can compute a γ-approximation $d' \in$

$[\Delta(\mathcal{C}_{\mathbf{A}}), \gamma \cdot \Delta(\mathcal{C}_{\mathbf{A}})]$ 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[3]. 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 return NO. Using binary search, one can efficiently find a d such that $O(\mathbf{A}, d) =$ YES and $O(\mathbf{A}, d-1) =$ NO. This means that $(\mathbf{A}, d)$ is *not* a NO instance and $(\mathbf{A}, d-1)$ is *not* a YES instance[4], and the minimum distance $\Delta(\mathcal{C}_{\mathbf{A}})$ 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 γ.

Definition 2 (Relatively Near Codeword Problem) *For prime power q, $\rho > 0$ and $\gamma \geq 1$, an instance of $\text{GAPRNC}^{(\rho)}_{\gamma,q}$ is a triple $(\mathbf{A}, \mathbf{v}, t)$, $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, $\mathbf{v} \in \mathbb{F}_q^n$ and $t \in \mathbb{Z}^+$, such that $t < \rho \cdot \Delta(\mathcal{C}_{\mathbf{A}})$ and*[5]

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

It is immediate that the problem $\text{RNC}^{(\rho)}$ gets harder as ρ increases. It is hardest when $\rho = \infty$ in which case we obtain the promise problem associated to approximating 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 $(\mathbf{A}, \mathbf{v}, t)$, $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, $\mathbf{v} \in \mathbb{F}_q^n$ and $t \in \mathbb{Z}^+$, such that*

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

The promise problem $\text{GAPNCP}_{\gamma,q}$ is NP-hard for every constant $\gamma \geq 1$ (cf. [2][6]), and this result is critical to our hardness result(s).

3. Hardness of the relatively near codeword problem

As outlined in Section 1, our reduction relies on the construction of a linear code $\mathcal{C}_{\mathbf{A}}$ and a Hamming sphere of radius $r < \rho \cdot \Delta(\mathcal{C}_{\mathbf{A}})$ (for some $\rho < 1$) containing exponentially (in the block length) many codewords. Obviously, it must be $\rho \geq \frac{1}{2}$ because any sphere of radius $r < \Delta(\mathcal{C}_{\mathbf{A}})/2$ can contain at most one codeword. We now prove that for any $\rho > \frac{1}{2}$ it is actually possible to build such a code and sphere. After the development of this combinatorial tool, we prove the hardness of approximating the relatively near codeword problem by reduction from the nearest codeword problem.

[3] By definition, when the input does not satisfies the promise, the oracle can return any answer.

[4] Remember that the oracle can give any answer if the input is neither a YES instance nor a NO one. So, one it would be wrong to conclude that $(\mathbf{A}, d-1)$ is a NO instance and $(\mathbf{A}, d)$ is a YES one.

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

[6] To be precise, Arora et al. [2] present the result only for binary codes. In fact, their proof is valid for any alphabet.

3.1. Construction of the combinatorial tool

We first show how to construct a linear code and a sphere (with radius smaller than the minimum distance of the code) containing a number of codewords exponential in the alphabet size. Then, we use code concatenation to derive a similar result for fixed alphabet in which the number of codewords in the sphere is exponential in the block length of the code.

Lemma 4 *For any $\epsilon \in (0,1)$, there exists an algorithm that, on input a prime power q, outputs, in $\text{poly}(q)$ time, three integers $l, m, r > 0$ and a matrix $\mathbf{A} \in \mathbb{F}_q^{l \times m}$ such that*

- *the linear code defined by $\mathbf{A}$ has minimum distance $\Delta(\mathcal{C}_{\mathbf{A}}) > 2(1-\epsilon)r$,*
- *the expected number of codewords inside a random sphere $\mathcal{B}(\mathbf{v}, r)$ ($\mathbf{v}$ chosen uniformly at random from $\mathbb{F}_q^l$) is at least $q^{\epsilon \lfloor q^\epsilon \rfloor}/4$.*

Proof: Let $r = \lfloor q^\epsilon \rfloor$, $l = q$ and $m = q - \lfloor 2(1-\epsilon)r \rfloor$. We let $\mathbf{A}$ be a generating matrix of the $[q, m, q-m+1]$ extended Reed-Solomon code (cf. [5, 12]). For example, let the columns of $\mathbf{A}$ correspond to the polynomials x^i (for $i = 0, \ldots, m-1$) evaluated on all elements of $\mathbb{F}_q$.

Clearly, $\mathbf{A}$ can be constructed in time polynomial in q and the minimum distance satisfies

$$\Delta(\mathcal{C}_{\mathbf{A}}) = q - m + 1 = \lfloor 2(1-\epsilon)r \rfloor + 1 > 2(1-\epsilon)r.$$

Now, lets bound the expected number of codewords in $\mathcal{B}(\mathbf{v}, r)$ when $\mathbf{v}$ is chosen uniformly at random in $\mathbb{F}_q^q$. First of all notice that

$$\begin{aligned}
\operatorname*{Exp}_{\mathbf{v} \in \mathbb{F}_q^q} [|\mathcal{C}_{\mathbf{A}} \cap \mathcal{B}(\mathbf{v}, r)|] &= \sum_{\mathbf{x} \in \mathcal{C}_{\mathbf{A}}} \Pr_{\mathbf{v} \in \mathbb{F}_q^q} \{\mathbf{x} \in \mathcal{B}(\mathbf{v}, r)\} \\
&= \sum_{\mathbf{x} \in \mathcal{C}_{\mathbf{A}}} \Pr_{\mathbf{v} \in \mathbb{F}_q^q} \{\mathbf{v} \in \mathcal{B}(\mathbf{x}, r)\} \\
&= \frac{|\mathcal{C}_{\mathbf{A}}| \cdot |\mathcal{B}(\mathbf{0}, r)|}{q^q} \\
&= q^{m-q} \cdot |\mathcal{B}(\mathbf{0}, r)| \\
&\geq q^{-2(1-\epsilon)r} \cdot |\mathcal{B}(\mathbf{0}, r)|.
\end{aligned}$$

Let's now bound the size of the ball $\mathcal{B}(\mathbf{0}, r)$:

$$|\mathcal{B}(\mathbf{0}, r)| \geq \binom{q}{r} (q-1)^r$$

$$
\begin{aligned}
&\geq \left(\frac{q}{r}\right)^r (q-1)^r \\
&\geq q^{(1-\epsilon)r}(q-1)^r \\
&= q^{(2-\epsilon)r}\left(1-\frac{1}{q}\right)^r \\
&\geq q^{(2-\epsilon)r}\left(1-\frac{1}{q}\right)^q \\
&\geq \frac{q^{(2-\epsilon)r}}{4}
\end{aligned}
$$

where in the last inequality we have used the monotonicity of $(1-1/q)^q$ and $q \geq 2$. Finally, combining the two inequalities we get

$$
\begin{aligned}
\operatorname*{Exp}_{\mathbf{v}\in\mathbb{F}_q^q}[|\mathcal{C}_{\mathbf{A}} \cap \mathcal{B}(\mathbf{v},r)|] &\geq q^{-2(1-\epsilon)r} q^{(2-\epsilon)r}/4 \\
&= q^{\epsilon r}/4 = q^{\epsilon\lfloor q^\epsilon \rfloor}/4.
\end{aligned}
$$

□

¿From the previous lemma, it immediately follows that there exists a sphere $\mathcal{B}(\mathbf{v},r)$ containing at least $q^{\epsilon\lfloor q^\epsilon\rfloor}/4$ codewords from $\mathcal{C}_{\mathbf{A}}$. However, while the lemma asserts that $\mathbf{A}$ and r can be easily computed, it is not clear how to efficiently determine the center of the sphere. Let $\mu = \operatorname*{Exp}_{\mathbf{v}\in\mathbb{F}_q^l}[|\mathcal{B}(\mathbf{v},r)\cap\mathcal{C}_{\mathbf{A}}|]$ be the expected number of codewords in the sphere when the center is chosen uniformly at random from $\mathbb{F}_q^l$. It is fairly easy to find spheres containing a number of codewords not much bigger than μ. In fact, by Markov's inequality $\Pr_{\mathbf{v}\in\mathbb{F}_q l}(|\mathcal{B}(\mathbf{v},r)\cap\mathcal{C}_{\mathbf{A}}| > \alpha\mu) < 1/\alpha$ when $\mathbf{v}$ is chosen uniformly at random from $\mathbb{F}_q^q$. It turns out that if $\mathbf{v}$ is chosen uniformly at random from $\mathcal{B}(\mathbf{0},r)$ (instead of the whole $\mathbb{F}_q^l$), then a similar lower bound can be proved. Namely, $\Pr_{\mathbf{v}\in\mathcal{B}(\mathbf{0},r)}(|\mathcal{B}(\mathbf{v},r)\cap\mathcal{C}_{\mathbf{A}}| < \delta\mu) < \delta$. In fact, this is just a special case of the following quite general fact.

Fact 5 *Let G be a group, $H \subset G$ a subgroup and $S \subset G$ an arbitrary subset of G. Let μ be the expected size of $H \cap Sz$ when z is chosen uniformly at random from G (here Sz denotes the set $\{s\cdot z \mid s\in S\}$). Choose $x \in S^{-1} = \{s^{-1} \mid s\in S\}$ uniformly at random. Then for any $\delta \leq 1$,*

$$
\Pr_{x\in S^{-1}}\{|H\cap Sx| \leq \delta\mu\} \leq \delta.
$$

Proof: First, we compute the expectation

$$
\begin{aligned}
\mu &= \operatorname*{Exp}_{z\in G}[|H\cap Sz|] \\
&= \sum_{y\in H}\Pr_{z\in G}\{y\in Sz\} \\
&= \frac{|H|\cdot|S|}{|G|}.
\end{aligned}
$$

Now, pick $y \in H$ uniformly at random and independently from x (which is chosen uniformly at random from S^{-1}). We notice that

$$
\begin{aligned}
\Pr_{x,y}\{xy = z\} &= \frac{|\{x\in S^{-1}, y\in H\colon y = x^{-1}z\}|}{|S|\cdot|H|} \\
&= \frac{|H\cap Sz|}{|H|\cdot|S|}.
\end{aligned}
$$

Moreover, since H is a subgroup, $Hy = H$ and $|Sx\cap H| = |Sxy\cap Hy| = |S(xy)\cap H|$. Therefore, denoting by $I(z)$ be the indicator variable that equals 1 if $|Sz\cap H| \leq \delta\mu$ and 0 otherwise, we can write

$$
\begin{aligned}
\Pr_x\{|Sx\cap H| \leq \delta\mu\} &= \Pr_{x,y}\{|S(xy)\cap H| \leq \delta\mu\} \\
&= \sum_{z\in G}\Pr_{x,y}\{xy = z\}\cdot I(z) \\
&= \sum_{z\in G}\frac{|Sz\cap H|\cdot I(z)}{|H|\cdot|S|} \\
&\leq \frac{|G|\cdot\delta\mu}{|H|\cdot|S|} = \delta
\end{aligned}
$$

□

Applying Fact 5 on group $G = (\mathbb{F}_q^q, +)$, subgroup $H = (\mathcal{C}_{\mathbf{A}}, +)$ and the set $S = \mathcal{B}(\mathbf{0},r)$ we immediately get the following corollary to Lemma 4. Notice then that if we set $\mathbf{v} = x$ then $Sx = \mathbf{v} + \mathcal{B}(\mathbf{0},r) = \mathcal{B}(\mathbf{v},r)$.

Corollary 6 *For any $\epsilon \in (0,1)$ there exists a probabilistic algorithm that on input a prime power q, outputs in time polynomial in q integers $l,m,r > 0$, a matrix $\mathbf{A} \in \mathbb{F}_q^{l\times m}$ and a vector $\mathbf{v} \in \mathbb{F}_q^l$ such that*

- *$\mathbf{A}$ defines a linear code with minimum distance $\Delta(\mathcal{C}_{\mathbf{A}}) > 2(1-\epsilon)r$,*
- *for any $\delta \leq 1$, the probability that $|\mathcal{B}(\mathbf{v},r)\cap\mathcal{C}_{\mathbf{A}}|$ is smaller than $\delta q^{\epsilon\lfloor q^\epsilon\rfloor}$ is at most 4δ.*

It is important to notice that in the previous lemma one must use arbitrarily large alphabets in order to get arbitrarily many codewords in the ball. We would like to prove a similar result in which the alphabet size can be kept fixed and only the block length of the code increases. This can be easily accomplished using the standard construction of *concatenating codes* [8]. The idea is to apply Corollary 6 to a sufficiently large extension field $\mathbb{F}_{q^c}$ and then represent each element of $\mathbb{F}_{q^c}$ as a sequence of elements of $\mathbb{F}_q$.

Lemma 7 *For $\epsilon \in (0,1)$ and finite field $\mathbb{F}_q$, there exists a probabilistic polynomial time algorithm that on input integers $k,s \in \mathbb{Z}^+$, outputs, in $\operatorname{poly}(k,s)$ time, integers $l,m,r \in \mathbb{Z}^+$, a matrix $\mathbf{A} \in \mathbb{F}_q^{l\times m}$ and a vector $\mathbf{v} \in \mathbb{F}_q^l$ such that*

- *$\Delta(\mathcal{C}_{\mathbf{A}}) > 2(1-\epsilon)r$*

- *The probability that $\mathcal{B}(\mathbf{v}, r)$ contains less than q^k codewords is at most q^{-s}.*

Proof: Let c be an integer such that $\epsilon c \lfloor q^{\epsilon c} \rfloor \geq k + s + 2$ and $q' = q^c$ is polynomial in s and k. For example, let $c = \lceil \epsilon^{-1} \cdot \max\{\log_q(k + s + 2), 1\} \rceil$.

Apply Corollary 6 to prime power $q' = q^c$ to obtain integers l', m', r', matrix $\mathbf{A}' \in \mathbb{F}_{q'}^{l' \times m'}$ and vector $\mathbf{v}' \in \mathbb{F}_{q'}^{l'}$ such that $\Delta(\mathcal{C}_{\mathbf{A}'}) > 2(1-\epsilon)r'$, and for all $\delta \leq 1$ the probability that $|\mathcal{B}(\mathbf{v}', r') \cap \mathcal{C}_{\mathbf{A}'}|$ is smaller than $\delta \cdot (q')^{\epsilon \lfloor (q')^{\epsilon} \rfloor}$ is at most 4δ. In particular, when $\delta = q^{-(s+2)}$, with probability at least $1 - 4 \cdot q^{-(s+2)} \geq 1 - q^{-s}$ we have

$$|\mathcal{B}(\mathbf{v}', r') \cap \mathcal{C}_{\mathbf{A}'}| \geq q^{c\epsilon \lfloor q^{c\epsilon} \rfloor} / q^{s+2} \geq q^{k+s+2} / q^{s+2} = q^k.$$

So, the sphere $\mathcal{B}(\mathbf{v}', r')$ contains the required number of codewords with sufficiently high probability. It only remains to reduce the alphabet size from q^c to q. This can be done concatenating the code $\mathcal{C}_{\mathbf{A}}$ with a $[q^c, c, q^c - q^{c-1}]$ linear Hadamard code. Details follow.

Recall that $\mathbb{F}_{q^c}$ is a c-dimensional vector space over $\mathbb{F}_q$. Fix a basis $b_1, \ldots, b_c \in \mathbb{F}_{q^c}$ of $\mathbb{F}_{q^c}$ over $\mathbb{F}_q$ and let $\phi_i : \mathbb{F}_{q^c} \to \mathbb{F}_q$ be the coordinate functions such that $x = \sum_{i=1}^{c} \phi_i(x) b_i$. Notice that the ϕ_i's are linear, i.e., $\phi_i(ax + by) = a\phi_i(x) + b\phi_i(y)$ for all $a, b \in \mathbb{F}_q$ and $x, y \in \mathbb{F}_{q^c}$. For all $x \in \mathbb{F}_{q^c}$, let now $h(x)$ be the sequence of all $\mathbb{F}_q$-linear combinations of the $\phi_i(x)$,

$$h(x) = \left(\sum_{i=1}^{c} a_i \phi_i(x) \right)_{a_1, \ldots, a_c \in \mathbb{F}_q}$$

and extend h to $\mathbb{F}_{q^c}^{q^c}$ componentwise

$$\tilde{h}(x_1, \ldots, x_{q^c}) = h(x_1), h(x_2), \cdots, h(x_{q^c}).$$

Notice that $h : \mathbb{F}_{q^c} \to \mathbb{F}_q^{q^c}$ is linear, $h(0) = 0$ and $\mathrm{wt}(h(x)) = q^{c-1}(q-1)$ for all $x \neq 0$. Therefore $\mathrm{wt}(\tilde{h}(\mathbf{w})) = q^{c-1}(q-1) \cdot \mathrm{wt}(\mathbf{w})$ and $\Delta(\tilde{h}(\mathbf{w}_1), \tilde{h}(\mathbf{w}_2)) = q^{c-1}(q-1) \cdot \Delta(\mathbf{w}_1, \mathbf{w}_2)$. We now define $\mathcal{C}_{\mathbf{A}}$ as the concatenation of $\mathcal{C}_{\mathbf{A}'}$ and h, i.e.,

$$\mathcal{C}_{\mathbf{A}} = \tilde{h}(\mathcal{C}_{\mathbf{A}'}) = \{\tilde{h}(\mathbf{w}) : \mathbf{w} \in \mathcal{C}_{\mathbf{A}}\}.$$

Further, let $\mathbf{v} = \tilde{h}(\mathbf{v}')$, $r = q^{c-1}(q-1) \cdot r'$, $l = q^c \cdot l'$ and $m = c \cdot m'$. A generating matrix $\mathbf{A} \in \mathbb{F}_q^{l' \times m'}$ for $\mathcal{C}_{\mathbf{A}}$ can be easily obtained replacing each element a in $\mathbf{A}'$ by the corresponding matrix $[h(a \cdot b_1) \mid \ldots \mid h(a \cdot b_c)] \in \mathbb{F}_q^{q^c \times c}$.

We claim that these settings satisfy the requirements of the lemma. Notice first that since $\mathrm{wt}(\tilde{h}(\mathbf{w})) = q^{c-1}(q-1) \cdot \mathrm{wt}(\mathbf{w})$, we have $\Delta(\mathcal{C}_{\mathbf{A}}) = q^{c-1}(q-1) \cdot \Delta(\mathcal{C}_{\mathbf{A}'}) > 2(1-\epsilon)r$. Further, $|\mathcal{C}_{\mathbf{A}} \cap \mathcal{B}(\mathbf{v}, r)| = |\mathcal{C}_{\mathbf{A}'} \cap \mathcal{B}(\mathbf{v}', r')|$ and thus the probability that $\mathcal{B}(\mathbf{v}, r)$ contains fewer than q^k codewords is at most q^{-s}. □

In the next subsection we will use the codewords inside the ball $\mathcal{B}(\mathbf{v}, r)$ to represent the solutions to a 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 $\mathbf{T} \in \mathbb{F}_q^{k \times l}$ and a vector $\mathbf{y} \in \mathbb{F}_q^l$, let $\mathbf{T}(y) = \mathbf{T}\mathbf{y}$ denote the linear transformation from $\mathbb{F}_q^l$ to $\mathbb{F}_q^k$. Further, let $\mathbf{T}(S) = \{\mathbf{T}(y) \mid y \in S\}$.

Lemma 8 *For any $\epsilon \in (0, 1)$ and finite field $\mathbb{F}_q$ there exists a probabilistic polynomial time algorithm that on input $(1^k, 1^s)$ outputs integers l, m, r, matrices $\mathbf{A} \in \mathbb{F}_q^{l \times m}$, and $\mathbf{T} \in \mathbb{F}_q^{k \times l}$ and a vector $\mathbf{v} \in \mathbb{F}_q^l$ such that*

1. *$\Delta(\mathcal{C}_{\mathbf{A}}) > 2(1-\epsilon)r$.*
2. *$\mathbf{T}(\mathcal{B}(\mathbf{v}, r) \cap \mathcal{C}_{\mathbf{A}}) = \mathbb{F}_q^k$ with probability at least $1 - q^{-s}$.*

Proof: Run the algorithm of Lemma 7 on input $(1^{2k+s+1}, 1^{s+1})$. Let $l, m, r, \mathbf{A}, \mathbf{v}$ be the output of the algorithm and define $S = \mathcal{B}(\mathbf{v}, r) \cap \mathcal{C}_{\mathbf{A}}$. The first property directly follows from the previous lemma. Moreover, with probability at least $1 - q^{-(s+1)}$ we have $|S| \geq q^{2k+s+1}$. Choose $\mathbf{T} \in \mathbb{F}_q^{k \times l}$ uniformly at random. We want to prove that with very high probability $\mathbf{T}(S) = \mathbb{F}_q^k$. Choose a vector $\mathbf{t} \in \mathbb{F}_q^k$ at random and define a new function $\mathbf{T}'(\mathbf{y}) = \mathbf{T}\mathbf{y} + \mathbf{t}$. Clearly $\mathbf{T}'(S) = \mathbb{F}_q^k$ iff $\mathbf{T}(S) = \mathbb{F}_q^k$. Notice that the random variables $\mathbf{T}'\mathbf{y}$ ($\mathbf{y} \in S$) 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 $\mathbf{y} \in S$ such that $\mathbf{T}'\mathbf{y} = \mathbf{x}$. By linearity of expectation and pairwise independence of the $\mathbf{T}'\mathbf{y}$ we have $\mathrm{Exp}[N_{\mathbf{x}}] = |S|p$ and $\mathrm{Var}[N_{\mathbf{x}}] = |S|(p - p^2) < |S|p$. Applying Chebychev's inequality we get

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

Therefore, for any $\mathbf{x} \in \mathbb{F}_q^k$, the probability that $\mathbf{T}'\mathbf{y} \neq \mathbf{x}$ for every $\mathbf{y} \in S$ is at most $q^{-(k+s+1)}$. By union bound, with probability at least $1 - q^{-(s+1)}$, for every $\mathbf{x} \in \mathbb{F}_q^k$ there exists a vector $\mathbf{y} \in S$ such that $\mathbf{T}\mathbf{y} = \mathbf{x}$. Adding up the error probabilities, we find that with probability at least $1 - (q^{-(s+1)} + q^{-(s+1)}) \geq 1 - q^{-s}$, $\mathbf{T}(S) = \mathbb{F}_q^k$, proving the second property. □

3.2. The reduction

We can now prove the inapproximability of the relatively near codeword problem.

Theorem 9 *For any $\rho > 1/2$, $\gamma \geq 1$ and any finite field $\mathbb{F}_q$, $\text{GAPRNC}^{(\rho)}_{\gamma,q}$ 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: Let η be an integer strictly bigger than $1/(2\rho - 1)$ and let $\gamma' = (\eta + 1)\gamma$. We prove the hardness of $\text{GAPRNC}^{(\rho)}_{\gamma,q}$ by reduction from $\text{GAPNCP}_{\gamma',q}$.

Let $(\mathbf{C}', \mathbf{v}', t')$ be an instance of $\text{GAPNCP}_{\gamma',q}$ with $\mathbf{C}' \in \mathbb{F}_q^{n\times k}$. We want to map it to an instance $(\mathbf{C}, \mathbf{v}, t)$ of $\text{GAPRNC}^{(\rho)}_{\gamma,q}$. Invoking Lemma 8 on input $(1^k, 1^s)$ and $\epsilon = 1-(1+1/\eta)/(2\rho) \in (0,1)$, we obtain integers l, m, r, a generating matrix $\mathbf{A} \in \mathbb{F}_q^{l\times m}$ of a linear code with minimum distance $\Delta(\mathcal{C}_\mathbf{A}) > 2(1-\epsilon)r = ((1+1/\eta)/\rho)r$, a matrix $\mathbf{T} \in \mathbb{F}_q^{k\times l}$ and a vector $\mathbf{w} \in \mathbb{F}_q^l$ such that $\mathbf{T}(\mathcal{C}_\mathbf{A} \cap \mathcal{B}(\mathbf{w}, r)) = \mathbb{F}_q^k$ with probability at least $1 - q^{-s}$.

Notice that $\mathbf{C}'\mathbf{T}\mathbf{A} \in \mathbb{F}_q^{n\times m}$ defines a linear code whose codewords are a subset of $\mathcal{C}_{\mathbf{C}'}$. Define $\mathbf{C}$ by stacking up $\eta t'$ copies of $\mathbf{A}$ and r copies of $\mathbf{C}'\mathbf{T}\mathbf{A}$:

$$\mathbf{C} = \left[\begin{array}{c} \mathbf{A} \\ \vdots \\ \mathbf{A} \\ \mathbf{C}'\mathbf{T}\mathbf{A} \\ \vdots \\ \mathbf{C}'\mathbf{T}\mathbf{A} \end{array}\right] \begin{array}{l} \left.\begin{array}{c}\\ \\ \\ \end{array}\right\} \eta t' \\ \left.\begin{array}{c}\\ \\ \\ \end{array}\right\} r \end{array}$$

Define vector $\mathbf{v}$ as the concatenation of $\eta t'$ copies of $\mathbf{w}$ and r copies of $\mathbf{v}'$:

$$\mathbf{v} = \left[\begin{array}{c} \mathbf{w} \\ \vdots \\ \mathbf{w} \\ \mathbf{v}' \\ \vdots \\ \mathbf{v}' \end{array}\right] \begin{array}{l} \left.\begin{array}{c}\\ \\ \\ \end{array}\right\} \eta t' \\ \left.\begin{array}{c}\\ \\ \\ \end{array}\right\} r \end{array}$$

Finally, let $t = (\eta + 1)t'r$. The output of the reduction is $(\mathbf{C}, \mathbf{v}, t)$.

First notice that (regardless of the input instance $(\mathbf{C}', \mathbf{v}', t')$), we can establish $t \leq \rho \cdot \Delta(\mathcal{C}_\mathbf{C})$ as follows:

$$\begin{aligned} \rho \cdot \Delta(\mathcal{C}_\mathbf{C}) &\geq \rho\eta t' \cdot \Delta(\mathcal{C}_\mathbf{A}) \\ &> \rho\eta t' \left(\frac{1+1/\eta}{\rho}\right) r \\ &= (\eta+1)t'r = t. \end{aligned}$$

We now prove that if $(\mathbf{C}', \mathbf{v}', t')$ is a YES instance of $\text{GAPNCP}_{\gamma',q}$, then $(\mathbf{C}, \mathbf{v}, t)$ is a YES instance of $\text{GAPRNC}^{(\rho)}_{\gamma,q}$, and if $(\mathbf{C}', \mathbf{v}', t')$ is a NO instance of $\text{GAPNCP}_{\gamma',q}$, then $(\mathbf{C}, \mathbf{v}, t)$ is a NO instance of $\text{GAPRNC}^{(\rho)}_{\gamma,q}$.

Assume $(\mathbf{C}', \mathbf{v}', t')$ is a NO instance, i.e., the distance of $\mathbf{v}'$ from $\mathcal{C}_{\mathbf{C}'}$ is greater than $\gamma' t'$. For all $\mathbf{x} \in \mathbb{F}_q^m$ we have

$$\begin{aligned} \Delta(\mathbf{C}\mathbf{x}, \mathbf{v}) &\geq r \cdot \Delta((\mathbf{C}'\mathbf{T}\mathbf{A})\mathbf{x}, \mathbf{v}') \\ &\geq r \cdot \Delta(\mathbf{v}', \mathcal{C}_{\mathbf{C}'}) \\ &> r \cdot \gamma' t' \\ &= r(\eta+1)\gamma t' = \gamma t \end{aligned}$$

proving that $(\mathbf{C}, \mathbf{v}, t)$ is a NO instance. (Notice that NO instances get mapped to NO instances with probability 1, as required.)

Conversely, assume $(\mathbf{C}', \mathbf{v}', t')$ is a YES instance, i.e., there exists $\mathbf{x}$ such that $\Delta(\mathbf{C}'\mathbf{x}, \mathbf{v}') \leq t'$. Let $\mathbf{y} = \mathbf{A}\mathbf{z}$ be a codeword in $\mathcal{C}_\mathbf{A}$ such that $\Delta(\mathbf{y}, \mathbf{w}) \leq r$ and $\mathbf{T}\mathbf{y} = \mathbf{x}$. We know such a codeword will exists with probability at least $1 - q^{-s}$. In such a case, we have

$$\begin{aligned} \Delta(\mathbf{C}\mathbf{z}, \mathbf{v}) &= \eta t' \Delta(\mathbf{A}\mathbf{z}, \mathbf{w}) + r\Delta(\mathbf{C}'\mathbf{T}\mathbf{A}\mathbf{z}, \mathbf{v}') \\ &\leq \eta t' r + r t' = t, \end{aligned}$$

proving that $(\mathbf{C}, \mathbf{v}, t)$ is a YES instance. □

Remark 10 *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 $\mathcal{B}(\mathbf{w}, r)$ such that $\mathbf{T}\mathbf{y} = \mathbf{x}$. 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 the introduction, hardness under polynomial RUR-reductions easily implies the following corollary.

Corollary 11 *For any $\rho > 1/2$, $\gamma \geq 1$ and any finite field $\mathbb{F}_q$, $\text{GAPRNC}^{(\rho)}_{\gamma,q}$ is not in* RP *unless* NP = RP.

4. 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 $\text{GAPRNC}^{(\rho)}_{\gamma,q}$. Then we use direct product constructions to amplify the inapproximability factor to any constant and to any factor $2^{\log^{(1-\epsilon)} n}$ ($\epsilon > 0$).

4.1. 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 $\text{GAPRNC}^{(1/\gamma)}_{\gamma,q}$.

Lemma 12 . *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 $\text{GAPRNC}^{\gamma^{-1}}_{\gamma,q}$. Let $(\mathbf{C},\mathbf{v},t)$ be an instance of $\text{GAPRNC}^{\gamma^{-1}}_{\gamma,q}$ and assume without loss of generality that $\mathbf{v}$ does not belong to code generated by $\mathbf{C}$. (One can easily check whether $\mathbf{v} \in \mathcal{C}_{\mathbf{C}}$ by solving a system of linear equations. If $\mathbf{v} \in \mathcal{C}_{\mathbf{C}}$ then $\Delta(\mathbf{v},\mathcal{C}_{\mathbf{C}}) = 0$ and $(\mathbf{C},\mathbf{v},t)$ is a YES instance.) Define the matrix $\mathbf{C}' = [\mathbf{C}|\mathbf{v}]$. We now prove that if $(\mathbf{C},\mathbf{v},t)$ is a YES instance of $\text{GAPRNC}^{\gamma^{-1}}_{\gamma,q}$, then $(\mathbf{C}',t)$ is a YES instance of $\text{GAPDIST}_{\gamma,q}$, and if $(\mathbf{C},\mathbf{v},t)$ is a NO instance of $\text{GAPRNC}^{\gamma^{-1}}_{\gamma,q}$, then $(\mathbf{C}',t)$ is a NO instance of $\text{GAPDIST}_{\gamma,q}$. Notice that in either case $\Delta(\mathcal{C}_{\mathbf{C}}) > \gamma t$.

Assume $(\mathbf{C},\mathbf{v},t)$ is a YES instance, i.e., there exists a $\mathbf{x}$ such that $\Delta(\mathbf{C}\mathbf{x},\mathbf{v}) \leq d$. Then, $\mathbf{C}\mathbf{x}-\mathbf{v}$ is a non-zero vector of the code generated by $\mathbf{C}'$ of weight at most t.

Conversely, assume $(\mathbf{C},\mathbf{v},t)$ is a NO instance and let $\mathbf{y} = \mathbf{C}\mathbf{x} + w\mathbf{v}$ be any non-zero vector of $\mathbf{C}'$. If $w = 0$ then $\mathbf{y} = \mathbf{C}\mathbf{x}$ is a non-zero element of $\mathcal{C}_{\mathbf{A}}$ and therefore $\text{wt}(\mathbf{y}) > \gamma t$ (using the promise). On the other hand, if $w \neq 0$ then $\text{wt}(\mathbf{y}) = \text{wt}(\mathbf{C}(w^{-1}\mathbf{x}) - \mathbf{v}) > \gamma t$ as $\Delta(\mathbf{v},\mathcal{C}_(\mathbf{C})) > \gamma t$. □

4.2. 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 13 *For $i \in \{1,2\}$, let $\mathcal{C}_i$ be a linear code generated by $\mathbf{A}_i \in \mathbb{F}_q^{n_i \times k_i}$. Then the direct product of $\mathcal{C}_1$ and $\mathcal{C}_2$, denoted $\mathcal{C}_1 \otimes \mathcal{C}_2$ is a code over $\mathbb{F}_q$ of block length $n_1 n_2$ and dimension $k_1 k_2$ whose codewords, when expressed as matrices in $\mathbb{F}_q^{n_1 \times n_2}$, are the set $\{\mathbf{A}_1 X \mathbf{A}_2^T | X \in \mathbb{F}_q^{k_1 \times k_2}\}$. A generating matrix for the code $\mathcal{C}_1 \otimes \mathcal{C}_2$ can be easily defined as the matrix $\mathbf{A}_1 \otimes \mathbf{A}_2 \in \mathbb{F}_q^{n_1 n_2 \times k_1 k_2}$ whose columns (when expressed as matrices) are given by $\mathbf{A}_1^{(j_1)} \cdot (\mathbf{A}_2^{(j_2)})^T$ where $\mathbf{A}_i^{(j_i)}$ is the ith column of $\mathbf{A}_i$ and $j_i \in \{1,\ldots,k_i\}$ for $i \in \{0,1\}$.*

Notice that the codewords of $\mathcal{C}_1 \otimes \mathcal{C}_2$ are matrices whose columns are codewords of $\mathcal{C}_1$ and rows are codewords of $\mathcal{C}_2$. In our reduction we will need the following fundamental property of direct product codes.

Proposition 14 *[12] For linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of minimum distance d_1 and d_2, their direct product is a linear code of distance $d_1 d_2$.*

Proof: Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be the generators of $\mathcal{C}_1$ and $\mathcal{C}_2$. Consider two codewords $M_1 = \mathbf{A}_1 X_1 \mathbf{A}_2^T$ and $M_2 = \mathbf{A}_1 X_2 \mathbf{A}_2^T$ of $\mathcal{C}_1 \otimes \mathcal{C}_2$. For any $\lambda_1, \lambda_2 \in \mathbb{F}_q$, the matrix $\lambda_1 M_1 + \lambda_2 M_2$ is also a codeword of $\mathcal{C}_1 \otimes \mathcal{C}_2$ since it can be expressed as $\mathbf{A}_1(\lambda_1 X_1 + \lambda_2 X_2)\mathbf{A}_2^T$. Thus $\mathcal{C}_1 \otimes \mathcal{C}_2$ is linear.

We now show that for any non-zero matrix X, $\mathbf{A}_1 X \mathbf{A}_2$ has at least $d_1 d_2$ non-zero entries. Consider first the matrix $\mathbf{A}_1 X$. Since this matrix is non-zero, there must be some column which is non-zero. Since every column is a codeword from $\mathcal{C}_1$, this implies that this column must have at least d_1 non-zero entries. Thus $\mathbf{A}_1 X$ has at least d_1 non-zero rows. Now consider the matrix $(\mathbf{A}_1 X)\mathbf{A}_2^T$. At least d_1 rows of this matrix are non-zero and each must have at least d_2 non-zero entries. This completes the claim.

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

We can now prove the following theorem.

Theorem 15 *For every finite field $\mathbb{F}_q$ the following holds:*

- *For every real $\gamma > 1$, $\text{GAPDIST}_{\gamma,q}$ 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 γ_0 be such that $\text{GAPDIST}_{\gamma_0,q}$ is hard by Lemma 12. Given an instance $(\mathbf{A},d)$ of $\text{GAPDIST}_{\gamma_0,q}$, consider the instance $(\mathbf{A}^{\otimes l}, d^l)$ of $\text{GAPDIST}_{\gamma_0^l,q}$, where

$$\mathbf{A}^{\otimes l} = \underbrace{(\cdots((\mathbf{A}\otimes\mathbf{A})\otimes\mathbf{A})\cdots\otimes\mathbf{A})}_{l}$$

is a generator matrix of

$$\mathcal{C}_{\mathbf{A}}^{\otimes l} = \underbrace{(\cdots((\mathcal{C}_{\mathbf{A}}\otimes\mathcal{C}_{\mathbf{A}})\otimes\mathcal{C}_{\mathbf{A}})\cdots\otimes\mathcal{C}_{\mathbf{A}})}_{l}$$

for an integer parameter $l \in \mathbb{Z}^+$. By Proposition 14 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.

To show the second part, just set $\gamma_0 = 2$ and $l = \log^{\frac{1-\epsilon}{\epsilon}} n$ in the previous reduction. 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 16 *For every finite field $\mathbb{F}_q$ the following holds:*

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

5. 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 map YES instances to YES instances with high probability. (This is similar to the hardness proof for the shortest vector problem in [3, 13].)

An obvious question is whether it is possible to remove the randomization and make the reduction deterministic. We notice that our reduction (as well as the Ajtai-Micciancio ones for SVP) uses randomness in a very restricted way. Namely, the only part of the reduction where randomness is used is the proof of Lemma 8. The construction in the lemma depends only on the input size, and not 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 statement in Lemma 8 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[7]:

Corollary 17 *For every finite field $\mathbb{F}_q$ the following holds:*

- *For any $\rho > 1/2$, and $\gamma \geq 1$* $\text{GAPRNC}^{(\rho)}_{\gamma,q}$ *is hard for* NP *under non-uniform deterministic polynomial reductions.*
- *For every real $\gamma > 1$,* $\text{GAPDIST}_{\gamma,q}$ *is hard for* NP *under non-uniform deterministic polynomial reductions.*
- *For every $\epsilon > 0$ and $\gamma(n) = 2^{\log^{(1-\epsilon)} n}$,* $\text{GAPDIST}_{\gamma,q}$ *is hard for* NP *under non-uniform deterministic quasi-polynomial reductions.*

We notice also that a *uniform* deterministic construction satisfying the properties of Lemma 8 would immediately give a proper NP-hardness result (i.e. hardness under deterministic Karp reductions) for the relatively near codeword problem and the minimum distance problem.

[7] 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.

References

[1] L. Adleman, "Two Theorems on Random Polynomial Time", in *Proc. 19th Symposium on Foundations of Computer Science* 1978, pp. 75–83.

[2] S. Arora, L. Babai, J. Stern, Z. Sweedyk, "The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations", *Journal of Computer and System Sciences*, Vol. 54, 1997 pp. 317–331.

[3] M. Ajtai, "The Shortest Vector Problem is NP-Hard for Randomized Reductions", in *Proc. 30th Symposium on Theory of Computing* 1998, pp. 10–19.

[4] S. Arora, C. Lund, "Hardness of Approximations", in D. S. Hochbaum (ed.), *Approximation Algorithms for* NP*-Hard Problems*, PWS Publishing, 1997.

[5] R.E. Blahut, *Theory and Practice of Error Control Codes,* Addison-Wesley, Reading, Massachusetts, 1983.

[6] J. Bruck, M. Naor, "The Hardness of Decoding Linear Codes with Preprocessing," *IEEE Transactions on Information Theory*, Vol. IT-36, n. 2, March 1990, pp. 381–385.

[7] E.R. Berlekamp, R.J. McEliece, H.C.A. van Tilborg, "On the Inherent Intractability of Certain Coding Problems", *IEEE Transactions on Information Theory*, Vol. IT-24, n. 3, May 1978, pp. 384–386.

[8] G.D. Forney, *Concatenated Codes*, Research Monograph No. 37, The MIT Press, Cambridge, Massachusetts, 1966.

[9] K. Jain, M. Sudan, V.V. Vazirani, Personal communication, May 1998.

[10] D. S. Johnson, "A Catalog of Complexity Classes", Chapter 2 in J. van Leeuwen (ed.) *Handbook of theoretical computer science* Vol. A (Algorithms and Complexity), Elsevier Science, 1990.

[11] J. Justesen, T. Høholdt, "Bounds on List Decoding of MDS codes", Manuscript, March 1999.

[12] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1981.

[13] D. Micciancio, "The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant", in *Proc. 39th Symposium on Foundations of Computer Science* 1998, pp. 92–98.

[14] M.A. Tsfasman and S.G. Vladuts, *Algebraic - Geometric Codes.* Dordrecht: Kluwer, 1991.

[15] A. Vardy. "The Intractability of Computing the Minimum Distance of a Code," *IEEE Trans. Inform. Theory*, Vol. IT-43, no. 6, November 1997, pp. 1757–1766.

[16] V. Zinoviev and S. Litsyn, "On codes exceeding the Gilbert bound," *Problems of Information Transmission.* Vol. 21, no. 1, 1985, pp. 109–111 (in Russian).