Low-Density Parity-Check Codes: Constructions and Bounds

Vitaly Skachek

Low-Density Parity-Check Codes: Constructions and Bounds

Research Thesis

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Vitaly Skachek

Submitted to the Senate of
the Technion — Israel Institute of Technology
Tevet 5766 Haifa January 2007

The research thesis was done under the supervision of Prof. Ronny Roth in the Computer Science Department.

Acknowledgements

Many people helped me in various ways in my work on this thesis. I am using this opportunity to thank them all.

First, I would like to express my gratitude to Ronny Roth, my Ph.D. advisor, for all his guidance, support and patience. I learned a lot from Ronny over this period.

I would like to thank Tuvi Etzion for his help when he temporarily replaced Ronny as my advisor. I benefited also from advices and stimulating discussions with Reuven Bar-Yehuda, Yuval Ishai, Simon Litsyn, Igal Sason, Amir Shpilka and Alexander Vardy. The administrative staff of the Computer Science Department was very helpful.

In 2004, I spent a very saturated and interesting summer at the Bell Laboratories of Lucent Technologies. I am especially thankful to Alexei Ashikhmin for organizing my visit and for a fruitful mentorship. I wish to thank people at the Bell Labs for their hospitality. Additionally, I would like to thank Jon Feldman of Columbia University for interesting discussions.

My interest in mathematics was significantly stimulated and developed during my studies at the 27-th Kharkiv Mathematical School back in the late 80's. I am using this opportunity to thank my math teacher at that time, Nelya Naumovna Strelchenko, for her selfless work. I am also thankful to the graduate students of the Kharkiv State University at that time, who volunteered to teach us, high school students, the smarts of math: Svetlana Eremenko, Alexander Kravchenko, Yakov Persky, Mikhail Simbirsky, Alexander Sokhet and Vitaly Ushakov.

I wish to thank my office-mates and fellow graduate students at the Technion for their assistance and support: Omer Barkol, Yael Ben-Haim, Ilya Blayvas, Alexander Brook, Zvi Devir, Bella Dubrov, Evgeniy Gabrilovich, Dan Kenigsberg, Alexander Landau, Beniamin Mounits, Vitaly Rubinovich, Sigal Saar, Ido Tal and others. I am also thankful to my friends for making my life interesting and enjoyable: Vitaly Braude, Michael Epstein, Alexander Feigel, Natalie Freidman, Kathy Ginzburg, Evgeni Koifman, Alexander and Fanya Nisenboim, Natalie Rublinetsky, Igor Weinstein, Naum Yoresh and others.

I am especially thankful to Inna Klyanfer.

Finally, I would like to thank those whom I owe the most: my parents, Inna Chertkov and Boris Skachek, for their love and support.

The generous financial support of the Joseph Freed and Benjamin Werber Fellowship, the Technion, and the DIMACS Special Focus Program in Computational Information Theory and Coding is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract

Low-density parity-check (LDPC) codes were introduced in 1962, but were almost forgotten. The introduction of turbo-codes in 1993 was a real breakthrough in communication theory and practice, due to their practical effectiveness. Subsequently, the connections between LDPC and turbo codes were considered, and it was shown that the latter can be described in the framework of LDPC codes. In recent years, low-density parity-check codes have been a subject of much experimentation and analysis. It was shown that, in practice, LDPC codes perform extremely well.

The most common approach to the analysis of LDPC codes is based on probabilistic methods, which consider so-called 'average LDPC codes'. This approach led to remarkable results on the performance of LDPC codes. However, there is still a gap between our understanding of LDPC code ensembles and characteristics of specific LDPC codes.

One promising approach for constructing specific LDPC codes is based on using *expander graphs*. Some of these codes (called *expander codes*) allow both linear-time encoding and decoding. Recently, it was shown that such codes can attain capacity of a variety of channels, while the error probability decreases exponentially with the code length. One of the main characteristics of any code family is a trade-off between its rate and its relative minimum distance: it was recently shown for the binary expander codes that this trade-off surpasses *the Zyablov bound*, which was used as a benchmark for evaluating the parameters of codes for many years.

In this thesis, we present improvements on the known bounds on the parameters of expander codes. We (slightly) improve the lower bound on the minimum distance of expander codes. We show that these codes can be viewed as a concatenation of a nearly-MDS expander code with an appropriate inner code. We suggest that GMD-decoding can be efficiently used for these codes. Thus, those nearly-MDS codes admit a linear-time encoding and decoding. Their alphabet size is smaller than the alphabet size of similar known codes.

By employing this approach, we are able to present a new decoding algorithm for expander codes together with a novel analysis. We show that our algorithm can correct (slightly) more errors than any known decoding algorithm for expander codes. Moreover, the decoding time of our algorithm has only polynomial dependence on the degree of the underlying graph.

Further, we investigate the decoding error probability of codes as a function of their block length. We show that the existence of codes with polynomially small decoding error probability implies the existence of codes with exponentially small decoding error probability. Specifically, we assume that there exists a family of codes of length N and rate $\mathcal{R} = (1-\varepsilon)\mathsf{C}$ (where C is the capacity of a binary symmetric channel), whose decoding probability decreases inverse polynomially in N. Then, we show that if the decoding probability decreases sufficiently fast, but still only inverse polynomially fast in N, then there exists another such family of codes whose decoding error probability decreases exponentially fast in N. Moreover, if the decoding time complexity of the assumed family of codes is polynomial in N and $1/\varepsilon$, then the decoding time complexity of the presented family is linear in N and polynomial in $1/\varepsilon$. We compare these codes to several known expander code families and show that the latter families cannot be tuned to having all aforementioned properties.

We construct so-called *generalized expander codes*, which are different from all known expander codes. We show that these generalized expander codes are (asymptotically) at least as good as the best known expander codes. We present a linear-time decoding algorithm for generalized expander codes.

We also consider expander codes defined over non-bipartite graphs and present a reduction from these codes to codes defined over bipartite graphs. This reduction leads to an efficient decoder for the former code family.

We also investigate expander codes that contain 'weak' constituent codes, i.e. constituent codes with a rather small minimum distance. We find lower and upper bounds on the minimum distance of an expander code having codes with minimum distance 2 as the constituent codes. In that case, we show that the overall code cannot be asymptotically good. Then, we derive some new lower bounds on the minimum distance of expander codes. Finally, we derive some sufficient conditions on the parameters of the constituent codes, such that the overall expander code family becomes asymptotically good.

Abbreviations and Notations

LDPC	Low-Density Parity-Check
IRA	Irregular Repeat Accumulative
MDS	Maximum Distance Separable
RS	Reed-Solomon
GRS	Generalized Reed-Solomon
BEC	Binary Erasure Channel
BSC	Binary Symmetric Channel
MP	Message Passing
BP	Belief Propagation
GF(q)	Galois Field of size q

$\lfloor x \rfloor$	Largest integer less than or equal to x.
$\lceil x \rceil$	Smallest integer greater than or equal to x.
$x \approx y$	x is approximately equal to y.
$A \cup B$	Union of the sets A and B.
$A \cap B$	Intersection of the sets A and B.
$A \backslash B$	Set difference of the sets A and B.
$\|A\|$	Number of elements in the set A.
$a \in A$	Element a is contained in the set A.
$B \subseteq A$	Set B is a subset of the set A or equals the set A.
$S_1 \Rightarrow S_2$	Statement S_1 implies S_2.
$S_1 \Leftrightarrow S_2$	Statement S_1 implies S_2 and S_2 implies S_1.
X^T	Transpose of the matrix X.
$\mathsf{w}(\boldsymbol{c})$	Relative Hamming weight of the word $\boldsymbol{c}$.
$\mathsf{w}_\mathsf{b}(\boldsymbol{c})$	Relative binary Hamming weight of the word $\boldsymbol{c}$.
$\mathsf{d}(\boldsymbol{x}, \boldsymbol{y})$	Hamming distance between the words $\boldsymbol{x}$ and $\boldsymbol{y}$.
$\mathsf{d}_2(\boldsymbol{x}, \boldsymbol{y})$	Binary Hamming distance between the words $\boldsymbol{x}$ and $\boldsymbol{y}$.

$\mathsf{E}(x)$	Expected value of the random variable x.
$\mathsf{Prob}(x)$	Probability of the event x.
$\mathsf{Prob}(y\vert x)$	Conditional probability of y given x.
$\mathsf{Prob}_e(\mathcal{C})$	Decoding error probability for the code $\mathcal{C}$ with respect to a prescribed decoder.
$\mathsf{P}_e(n)$	Average decoding error probability of a random linear code of length n with respect to the maximum-likelihood decoder.
C	Capacity of a channel.
$\mathsf{C}_q(p)$	Capacity of the q-ary symmetric channel with crossover probability p.
$\mathsf{H}_q(p)$	q-ary entropy function.
α	Relative size of expanding set in an expander graph.
ζ	Expansion factor of an expander graph.
$g(\mathcal{G})$	Girth of the graph $\mathcal{G}$.
$A_{\mathcal{G}}$	Adjacency matrix of the graph $\mathcal{G}$.
λ	Second largest eigenvalue of the matrix $A_{\mathcal{G}}$ (where $\mathcal{G}$ is regular).
λ^*	Second largest absolute value of eigenvalue of the matrix $A_{\mathcal{G}}$ (where $\mathcal{G}$ is regular).
Δ	Degree of a vertex in a regular expander graph.
$\gamma_{\mathcal{G}}$	Ratio between λ and Δ for the matrix $A_{\mathcal{G}}$ (where $\mathcal{G}$ is regular).
$\gamma_{\mathcal{G}}^*$	Ratio between λ^* and Δ for the matrix $A_{\mathcal{G}}$ (where $\mathcal{G}$ is regular).
$\mathcal{E}$	Encoder.
$\mathcal{D}$	Decoder.
$\mathbb{E}(r)$	Random coding exponent.
$\mathcal{M}$	Message alphabet of a message-passing decoder.
$\mathcal{O}$	Output alphabet of a channel.
$E(v)$	Set of edges incident with the vertex v.
$\mathcal{G}_S$	Graph induced from the graph $\mathcal{G}$ by the vertex set S.
E_S	Set of edges induced by the vertex set S.
$\mathcal{N}(v)$	Set of neighbors of the vertex v.
$\deg(v)$	Degree of the vertex v in a graph.
$\deg_S(v)$	Degree of vertex v in the subgraph induced by the vertex set S.
$(\boldsymbol{x})_{E(v)}$	Sub-word of $\boldsymbol{x}$ indexed by $E(v)$.
$\mathbb{C} = (\mathcal{G}, \mathcal{C})$	Low-complexity code with an underlying graph $\mathcal{G}$ and a constituent code $\mathcal{C}$.
$\mathcal{C}\left[\mathcal{R}, N\right]$	Code $\mathcal{C}$ of rate $\mathcal{R}$ and length N.
$\delta_Z(\mathcal{R})$	The Zyablov bound for binary codes of rate $\mathcal{R}$.
$\delta_{GV}(\mathcal{R})$	Gilbert-Varshamov bound for binary codes of rate $\mathcal{R}$.

Chapter 1

Introduction

1.1 Background

Low-density parity-check (in short, LDPC) codes were first introduced by Gallager in 1962 [33]. In his pioneering work, Gallager introduced iterative decoding algorithms for LDPC codes. He showed that for all code rates below a certain bound (which is lower than the Shannon capacity), the decoding error probability for these algorithms decays exponentially with the square root of the code length. However, Gallager's work was almost forgotten for several decades.

The introduction of turbo codes in 1993 by Berrou, Glavieux, and Thitimajshima [12], was a real breakthrough in communication theory and practice, due to their practical effectiveness. The technique of so-called iterative decoding was shown empirically to perform at rates closer to the Shannon capacity of the channel than any known algebraic decoder could do. However, as of yet, there is no full analysis of the performance of turbo codes.

Later on, the connection between LDPC codes and turbo codes was considered. It was shown that the latter can be described in the framework of low-density parity-check codes (see for example, [57]). Moreover, the turbo decoding algorithm [49] can be understood as a belief propagation algorithm [63]. Hence, belief propagation analysis that is performed on LDPC codes may be applicable to turbo codes as well.

In the last few years, LDPC codes have been the subject of much experimentation and analysis. In the present research, we further study these codes. Specifically, in this thesis, we present some results on the achievable parameters of such codes. We improve further on the existing LDPC code constructions and algorithms, in particular, by constructing new codes with better parameters than the existing ones, and by developing new, more efficient, encoding-decoding algorithms for new and existing codes.

The contents of this thesis is as follows. In Chapter 1, we present the definitions that will be used throughout this document and survey the key results that have been published on LDPC codes. In Chapters 2 through 6 we present the results of our research. Finally, in Chapter 7, we summarize our results and discuss related open problems.

1.2 Definitions

1.2.1 Basic definitions

A finite sequence of elements of an alphabet Σ is called a *word* over Σ. A set $\mathcal{C}$ of words of length n over the alphabet Σ is called a *code* over Σ, and n is called the length of the code $\mathcal{C}$.

Consider two words $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ in Σ^n. The *Hamming distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as the number of pairs of symbols (x_i, y_i), $1 \le i \le n$, such that $x_i \neq y_i$, and is denoted by $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y})$. The *minimum distance* of a code $\mathcal{C}$ is defined as

$$d = \min_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C},\ \boldsymbol{x} \neq \boldsymbol{y}} \mathsf{d}(\boldsymbol{x}, \boldsymbol{y}).$$

The *relative minimum distance* of $\mathcal{C}$ is defined as $\delta = d/n$, where n is the code length.

Denote by $\boldsymbol{x}^T$ the transpose of vector $\boldsymbol{x}$. A code $\mathcal{C}$ over a field $\mathbb{F}$ is said to be a *linear* $[n, k, d]$ *code* if there exists matrix H with n columns and rank $n-k$ such that for all $\boldsymbol{x} \in \mathbb{F}^n$

$$H\boldsymbol{x}^T = \overline{0} \;\Leftrightarrow\; \boldsymbol{x} \in \mathcal{C},$$

and the minimum distance of the code $\mathcal{C}$ is d. The matrix H is called a *parity-check matrix* of the code $\mathcal{C}$. The value k is called the *dimension* of the code $\mathcal{C}$, and the ratio $\mathcal{R} = k/n$ is called the *rate* of the code $\mathcal{C}$.

1.2.2 Channels and Shannon capacity

A *discrete memoryless communication channel* is defined by the following three ingredients:

1. Input alphabet $\{a_1, a_2, \ldots, a_I\}$.

2. Output alphabet $\{b_1, b_2, \cdots, b_J\}$.

3. Conditional probability assignment $\mathsf{Prob}(b_j | a_i)$ for each pair of symbols (a_i, b_j), $1 \le i \le I$, $1 \le j \le J$.

The memoryless condition induces the following conditional probability measure for every input word $(x_1, x_2, \ldots, x_n)$ and output word $(y_1, y_2, \ldots, y_n)$ (of the same length):

$$P\left\{y_1, y_2, \ldots, y_n \middle| x_1, x_2, \ldots, x_n\right\} = \prod_{i=1}^{n} \mathsf{Prob}(y_i|x_i).$$

An important special case of discrete memoryless communication channels is the *q-ary symmetric channel*, for which $I = J = q$ and

$$\forall i \in 1, \cdots, I, \ \forall j \in 1, \cdots, J, \ i \neq j \ : \mathsf{Prob}(b_j|a_i) = \frac{p}{q-1} \ ,$$

where p is a constant called the *crossover probability* of the channel and it does not depend on a_i or b_j. In particular, when $q = 2$, the channel is called the *binary symmetric channel* (in short, BSC).

Given the q-ary symmetric channel with crossover probability $p < 1 - \frac{1}{q}$, the *Shannon capacity* of this channel is given by

$$\mathsf{C}_q(p) = 1 - \mathsf{H}_q(p),$$

where $\mathsf{H}_q : [0, 1] \longrightarrow [0, 1]$ is the q-ary entropy function

$$\mathsf{H}_q(x) = x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x),$$

with $\mathsf{H}_q(0) = 0$ and $\mathsf{H}_q(1) = \log_q(q-1)$. One of the main results in information theory is the Shannon Coding Theorem [78], which states that for any design rate $\mathcal{R} < \mathsf{C}_q(p)$ there exists an infinite family of codes $\{\mathcal{C}_i\}_{i=0}^{\infty}$ satisfying the following conditions:

- The length of $\mathcal{C}_i$ approaches infinity as $i \to \infty$.
- The rate $\mathcal{R}_i$ of $\mathcal{C}_i$ satisfies $\mathcal{R} < \mathcal{R}_i < \mathsf{C}_q(p)$ for all i.
- The decoding error probability under maximum-likelihood decoding of the codes in the family decays exponentially with the code length.

In the sequel, we say that the code family *attains the Shannon capacity* if it satisfies these three conditions. Shannon's proof of the coding theorem is non-constructive. The problem of finding explicit code constructions that attain the Shannon capacity was studied extensively over the years.

1.2.3 LDPC codes

Low-density parity-check (LDPC) codes over GF(2) are commonly described in terms of bipartite graphs. Let $\mathsf{G} = (\mathsf{V}, E_{\mathsf{G}})$ be a bipartite undirected graph with a vertex set $\mathsf{V} = \mathsf{V_m} \cup \mathsf{V_c}$ such that $\mathsf{V_m} \cap \mathsf{V_c} = \emptyset$ and an edge set E_{G} such that every edge in E_{G} has one endpoint in $\mathsf{V_m}$ and one endpoint in $\mathsf{V_c}$. In the following, we refer to the vertices in $\mathsf{V_m}$ and in $\mathsf{V_c}$ as *message* and *check* vertices, respectively. The graph G with $|\mathsf{V_m}| = n$ and $|\mathsf{V_c}| = r$ produces a linear code of block length n and dimension $k \geq n - r$ in the following way. The entries of a codeword are indexed by the message vertices. A vector $\boldsymbol{x} = (x_1, ..., x_n)$ is a codeword if and only if $H\boldsymbol{x}^T = 0$, where H is the $r \times n$ incidence matrix of the graph G: the rows of H are indexed by the check vertices and the columns of H are indexed by the message vertices. For each $i = 1, \ldots, n$ and $j = 1, \ldots, r$ the entry $(H)_{i,j}$ equals 1 if there is an edge between message vertex i and check vertex j in the bipartite graph G; otherwise $(H)_{i,j}$ equals 0. In other words, $\boldsymbol{x}$ is a codeword if and only if for each check vertex v, the sum of entries in $\boldsymbol{x}$ that are indexed by the message vertices adjacent to v is zero. In the common terminology of linear codes, the matrix H is a parity-check matrix of the LDPC code.

It follows from the above definitions that the degree of the message (check) vertex i is equal to the number of non-zero entries in the i'th column (row) of the matrix H. For LDPC codes the number of non-zero entries in each column (row) is typically bounded by a small constant. This explains the origin of the name *low-density* parity-check codes. The definition of LDPC codes can be extended toward codes over a general field $\mathbb{F} = \mathrm{GF}(q)$.

For *regular* LDPC codes the degrees of all message vertices are equal, and the degrees of all check vertices are equal. This means that the parity-check matrix H contains the same number of ones in each row and the same number of ones in each column. By contrast, *irregular* LDPC codes are codes that are based on graphs where the degree of the vertices on each side of the graph can vary.

1.2.4 Low-complexity codes

The following construction is due to Tanner [82]. Let $\mathcal{G} = (V, E)$ be a Δ-regular undirected graph with a vertex set V, and an edge set E of size $N = \frac{1}{2}|V|\Delta$. For every vertex $v \in V$, denote by $E(v)$ the set of edges incident with v. We assume an ordering on the edges of $E(v)$ for every $v \in V$, and let $\mathbb{F} = \mathrm{GF}(q)$. For a word $\boldsymbol{x} = (x_e)_{e \in E}$ (whose entries are indexed by E) in $\mathbb{F}^N$, denote by $(\boldsymbol{x})_{E(v)}$ the sub-word of $\boldsymbol{x}$ that is indexed by $E(v)$.

Fix $\mathcal{C}$ to be a linear $[\Delta, k{=}r\Delta, d]$ code over $\mathbb{F}$. The code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ is defined in [82] to be the following linear $[N, K, D]$ code over $\mathbb{F}$:

$$\mathbb{C} = \left\{ \boldsymbol{c} \in \mathbb{F}^N \; : \; (\boldsymbol{c})_{E(v)} \in \mathcal{C} \text{ for every } v \in V \right\} \; .$$

It was shown in [82] by Tanner that $K/N \geq 2r-1$. The code $\mathbb{C}$ is called a *low-complexity code* and the graph $\mathcal{G}$ is called a *Tanner graph.*

1.2.5 Expander codes

Consider a Δ-regular graph $\mathsf{G} = (\mathsf{V}, E_{\mathsf{G}})$ with a vertex set V and an edge set E_{G}. For every vertex $v \in \mathsf{V}$, denote by $\mathcal{N}(v)$ the set of vertices adjacent to v. We say that a subset $S \subseteq \mathsf{V}$ expands by a factor of ζ, $0 < \zeta \leq 1$, if

$$\left| \bigcup_{v \in S} \mathcal{N}(v) \right| \geq \zeta \Delta \cdot |S|.$$

We say that the graph G is an (α, ζ)-*expander* if every subset of at most $\alpha|\mathsf{V}|$ vertices expands by a factor of ζ.

Consider a graph G where each vertex has degree Δ.

Denote by A_{G} the adjacency matrix of G; namely, A_{G} is a $|\mathsf{V}| \times |\mathsf{V}|$ real symmetric matrix whose rows and columns are indexed by the set V, and for every $u, v \in \mathsf{V}$, the entry in A_{G} that is indexed by (u, v) is given by

$$(A_{\mathsf{G}})_{u,v} = \left\{ \begin{array}{ll} 1 & \text{if } \{u, v\} \in E_{\mathsf{G}} \\ 0 & \text{otherwise} \end{array} \right. .$$

It is easy to see that Δ is the largest eigenvalue of A_{G}. Let λ^* be the second largest absolute value of any eigenvalue of A_{G}. It was shown in [1] that lower ratios $\frac{\lambda^*}{\Delta}$ imply greater values ζ of expansion.

An expander graph for which the relation

$$\lambda^* \leq 2\sqrt{\Delta - 1} \tag{1.1}$$

holds is called a *Ramanujan graph.* Ramanujan graphs have essentially the smallest possible value of λ^* (given Δ) [1]. It is known that there exist infinite families of such graphs with the number of vertices approaching infinity for fixed values of vertex degree Δ [53], [62]. We denote by γ_{G} the ratio between the second largest eigenvalue of A_{G} and Δ (this ratio is less than 1 when G is connected and is nonnegative when $|\mathsf{V}| > 1$; see [22, Propositions 1.1.2 and 1.1.4]).

In the remaining part of this section we will discuss bipartite graphs. Take the bipartite graph $\mathsf{G} = (\mathsf{V}, E_{\mathsf{G}})$, with a vertex set $\mathsf{V} = \mathsf{V}_{\mathsf{m}} \cup \mathsf{V}_{\mathsf{c}}$, $\mathsf{V}_{\mathsf{m}} \cap \mathsf{V}_{\mathsf{c}} \neq \emptyset$, and an edge set E_{G}. We consider the expansion of sets of message vertices contained within the set V_{m} only. The bipartite graph G will be called an *unbalanced bipartite* (α, ζ)-*expander* if every subset of V_{m} of at most $\alpha|\mathsf{V}_{\mathsf{m}}|$ vertices, $\alpha < 1$, expands by a factor of at least ζ, $0 < \zeta < 1$.

Consider an unbalanced bipartite expander where every vertex in $\mathsf{V_m}$ has the same degree Δ_{m} and every vertex in $\mathsf{V_c}$ has the same degree Δ. We assume an ordering on the vertices of $\mathcal{N}(v)$ for every $v \in \mathsf{V_c}$. Let $\mathcal{C}$ be a linear error-correcting code of length Δ over $\mathbb{F} = \mathrm{GF}(q)$. The *expander code* $\mathbb{C} = \langle \mathsf{G}, \mathcal{C} \rangle$ is the code of length $|\mathsf{V_m}|$ over $\mathbb{F}$ whose codewords are all words $\boldsymbol{x} = (x_1, x_2, \ldots, x_{|\mathsf{V_m}|})$ over $\mathrm{GF}(q)$ such that, for every $v \in \mathsf{V_c}$, the sub-word of $\boldsymbol{x}$ that is indexed by $\mathcal{N}(v)$ is a codeword of $\mathcal{C}$.

Suppose now that $\Delta_{\mathsf{m}} = 2$. We define a new graph $\mathcal{G} = (V, E)$ with a vertex set V and an edge set E as follows:

$$\begin{array}{l} \bullet \;\; V = \mathsf{V_c} \\ \bullet \;\; \forall v, v' \in \mathsf{V_c}, v \neq v' \;:\; \{v, v'\} \in E \;\Leftrightarrow\; \mathcal{N}(v) \cap \mathcal{N}(v') \neq \emptyset \end{array} . \tag{1.2}$$

Note that the produced graph $\mathcal{G}$ is a Tanner graph and the code $\mathbb{C}$ is the low-complexity code $(\mathcal{G}, \mathcal{C})$. The entries of a codeword are now indexed by edges of the graph $\mathcal{G}$ rather than by vertices of G.

1.2.6 MDS codes and asymptotically good codes

A given linear $[n, k, d]$ code C over the field $\mathrm{GF}(q)$ is said to satisfy the *Singleton bound* if $n = k + d - 1$. The code that satisfies this condition is called *maximum distance separable* (MDS). There are several known MDS constructions. For example, the repetition code of length n over $\mathrm{GF}(q)$ that consists of all vectors of length n with all-the-same entries from $\mathrm{GF}(q)$ is one example of MDS code. Reed-Solomon (in short, RS) codes and generalized Reed-Solomon (in short, GRS) codes [59, Chapter 10] are non-trivial examples of such codes.

A family of codes $\{C_i\}_{i=0}^{\infty}$, where each C_i is a linear $[n_i, k_i, d_i]$ code, is said to be *asymptotically good* if it satisfies the following conditions:

- The length n_i of C_i approaches infinity as $i \to \infty$.
- $\lim_{i\to\infty} \frac{d_i}{n_i} = \delta > 0$
- $\lim_{i\to\infty} \frac{k_i}{n_i} = \mathcal{R} > 0$

1.2.7 Concatenated codes and Justesen codes

In this section, we revisit the definition of concatenated codes. The following ingredients will be used:

- A linear $[\mathsf{n}_{in}, k_{in}{=}R_{in}\mathsf{n}_{in}, \delta_{in}\mathsf{n}_{in}]$ code $\mathcal{C}_{in}$ over $\mathbb{F} = \mathrm{GF}(q)$ (inner code).

- A linear $[n, R_\Phi n, \delta_\Phi n]$ code $\mathbb{C}_\Phi$ over $\Phi = \mathbb{F}^{k_{in}}$ (outer code).
- A linear one-to-one mapping $\mathcal{E}_0 \; : \; \Phi \to \mathcal{C}_{in}$.

The respective concatenated code $\mathbb{C}_{cont}$ of length $n \cdot \mathsf{n}_{in}$ over $\mathbb{F}$ is defined as

$$\begin{aligned} \mathbb{C}_{cont} = \Big\{ & (\boldsymbol{c}_1 | \boldsymbol{c}_2 | \cdots | \boldsymbol{c}_n) \in \mathbb{F}^{n \cdot \mathsf{n}_{in}} : \boldsymbol{c}_i = \mathcal{E}_0(\Xi_i) \;, \\ & \text{for } i \in 1, 2, \cdots, n, \;\; \text{and } (\Xi_1 \Xi_2 \cdots \Xi_n) \in \mathbb{C}_\Phi \Big\} \;. \end{aligned}$$

The rate of $\mathbb{C}_{cont}$ is known to be $R_{cont} = R_{in} \cdot R_\Phi$. The relative minimum distance of $\mathbb{C}_{cont}$, δ, is at least $\delta \geq \delta_{in} \cdot \delta_\Phi$.

Let $\mathcal{D}_{in} : \mathbb{F}^{\mathsf{n}_{in}} \longrightarrow \mathcal{C}_{in}$ and $\mathcal{D}_\Phi : \Phi^n \longrightarrow \mathbb{C}_\Phi$ be decoders for the codes $\mathcal{C}_{in}$ and $\mathbb{C}_\Phi$, respectively. A simple decoder $\mathcal{D}_{cont}$ for the code $\mathbb{C}_{cont}$ is presented in Figure 1.1. This decoder is able to correct any error pattern of less then $\frac{1}{4}\delta_{in}\delta_\Phi$ errors over $\mathbb{F}$.

```
Input:   received word $\boldsymbol{y} = (y_1 \, y_2 \, \cdots \, y_{n \cdot \mathsf{n}_{in}})$ in $\mathbb{F}^{n \cdot \mathsf{n}_{in}}$.
For $i = 1, 2, \cdots, n$   do   $u_i \leftarrow \mathcal{E}_0^{-1}\left(\mathcal{D}_{in}\left( \, (y_{j+(i-1)\cdot \mathsf{n}_{in}})_{j=1}^{\mathsf{n}_{in}} \, \right)\right)$.
Let $(z_1 z_2 \cdots z_n) \leftarrow \mathcal{D}_\Phi\left((u_1 u_2 \cdots u_n)\right)$.
Output:   $(\mathcal{E}_0(z_1) | \mathcal{E}_0(z_2) | \cdots | \mathcal{E}_0(z_n))$.
```

Figure 1.1: Decoder $\mathcal{D}_{cont}$ for the concatenated code $\mathbb{C}_{cont}$.

Concatenated codes were invented by Forney [30], [31]. In his works, Forney proposed a polynomial-time decoding technique called *generalized minimum distance decoder* (in short, GMD decoder). The GMD decoder $\mathcal{D}_{\text{GMD}}$ is presented in Figure 1.2.

When the output of the decoder $\mathcal{D}_{in}$ is suspected to be unreliable, the decoder $\mathcal{D}_{\text{GMD}}$ uses a special symbol '?' (erasure) instead of the actual output of $\mathcal{D}_{in}$. The value h_t is used as a varying threshold of reliability for $\mathcal{D}_{in}$. The decoder $\mathcal{D}_{\text{GMD}}$ makes use of the error-erasure decoder $\mathcal{D}'_\Phi : (\Phi \cup \{\text{'?'}\})^n \longrightarrow \mathbb{C}_\Phi$ (for the code $\mathbb{C}_\Phi$), which corrects any pattern of ϑ errors and ρ erasures, given that $\vartheta + \rho/2 < \delta_\Phi n/2$. The decoder $\mathcal{D}_{\text{GMD}}$ therein is able to correct any error pattern of less then $\frac{1}{2}\delta_{in}\delta_\Phi$ errors over $\mathbb{F}$ (see [30] for details).

For any given design rate $\mathcal{R} < \mathsf{C}_q(p)$, Forney constructed a series of concatenated codes $\{C_i\}_{i=1}^\infty$, where each C_i has length N_i and rate R_i, such that

- The rate R_i satisfies $\mathcal{R} \leq R_i$.

```
Input:  received word $\boldsymbol{y} = (y_1\, y_2\, \cdots\, y_{n\cdot\mathsf{n}_{in}})$ in $\mathbb{F}^{n\cdot\mathsf{n}_{in}}$.

For $i \in 1, 2, \cdots, n$  do  {
        Let  $\boldsymbol{x}_i \leftarrow \mathcal{D}_{in}\big( (y_{j+(i-1)\cdot\mathsf{n}_{in}})_{j=1}^{\mathsf{n}_{in}} \big)$  .
        Let  $u_i \leftarrow \mathcal{E}_0^{-1}(\boldsymbol{x}_i)$  .
    }

For $h_t = 1, 2, \cdots, \delta_{in}\mathsf{n}_{in}$  do  {
        For $i \in 1, 2, \cdots, n$  let  $w_i \leftarrow \begin{cases} u_i & \text{if } \mathsf{d}(\boldsymbol{x}_i, (y_{j+(i-1)\cdot\mathsf{n}_{in}})_{j=1}^{\mathsf{n}_{in}}) < h_t \\ ? & \text{otherwise} \end{cases}$  .
        Let $(z_1 z_2 \cdots z_n) \leftarrow \mathcal{D}'_{\Phi}((w_1 w_2 \cdots w_n))$  .
        Let  $\boldsymbol{a} \leftarrow (\mathcal{E}_0(z_1)|\mathcal{E}_0(z_2)|\cdots|\mathcal{E}_0(z_n))$  .
        If  $\mathsf{d}(\boldsymbol{a}, \boldsymbol{y}) < \frac{1}{2}\delta_{in}\mathsf{n}_{in} \cdot \delta_{\Phi} n$  then  output $\boldsymbol{a}$ and halt  .
    }

Output: 'failure to decode'.
```

Figure 1.2: Decoder $\mathcal{D}_{\text{GMD}}$ for the concatenated code $\mathbb{C}_{cont}$.

- The decoding error probability of the code C_i under the GMD decoder is bounded from above by
$$\mathsf{Prob}_e(C_i) \leq \max_{\mathcal{R} \leq r \leq \mathsf{C}_q(p)} \mathsf{e}^{-N_i \mathbb{E}(r)(1-\frac{\mathcal{R}}{r})}, \tag{1.3}$$
where $\mathbb{E}(r)$ is the so-called *random coding exponent* defined in [35] as
$$\mathbb{E}(r) = \lim_{\mathsf{n}\to\infty} \sup \frac{-\ln \mathsf{P}_e(\mathsf{n})}{\mathsf{n}}, \tag{1.4}$$
and $\mathsf{P}_e(\mathsf{n})$ is the average decoding error probability of a random linear code of length n and rate r under the maximum-likelihood decoder, as defined in [35, p. 121].

In Forney's work, the inner code C_{in} was taken to be a block code of length n_{in} and rate R_{in} that has the smallest decoding error probability. The outer code C_{out} was taken to be a GRS code. Forney showed that this concatenated code construction attains the Shannon capacity. However, the decoding in Forney's work has super-linear time.

Let $z \mapsto \mathsf{H}_q^{-1}(z)$ be the inverse of the q-ary entropy function $x \mapsto \mathsf{H}_q(x)$ over the interval $[0, 1-1/q]$. A modified version of concatenated codes was proposed by Justesen in [41]. The proposed codes were proved to satisfy the inequality
$$\delta \geq \max_{\max\{\mathcal{R},1/2\} \leq r \leq 1} \left(1 - \frac{\mathcal{R}}{r}\right) \mathsf{H}_q^{-1}(1-r).$$

Therefore, for a certain interval of rates, Justesen codes attain the Zyablov bound

$$\delta \geq \max_{\mathcal{R} \leq r \leq 1} \left(1 - \frac{\mathcal{R}}{r}\right) \mathsf{H}_q^{-1}(1-r). \tag{1.5}$$

1.2.8 Message-passing decoders

In several works on LPDC codes, the analysis of decoding is done under the *message-passing* (in short, MP) algorithms. The first message-passing algorithms for decoding of LDPC codes were proposed by Gallager in [34]. Some of the message-passing algorithms, including one of the algorithms proposed by Gallager, are so-called *belief-propagation* (BP) algorithms [66]. Belief-propagation algorithms were extensively used and investigated by researchers in the field of artificial intelligence. The turbo decoding [12] was shown in [63] to be another instance of belief-propagation algorithms.

We next describe the structure of a message-passing decoder for LDPC codes.

- Suppose that G is a bipartite graph, consisting of message vertices $\{v_i\}$ and check vertices $\{c_j\}$.
- The decoder has a message alphabet $\mathcal{M}$. This alphabet is used for sending messages between the vertices of G, i.e. each message is one of the symbols of $\mathcal{M}$.
- Denote by $\mathcal{O}$ the output alphabet of the channel. Assume $\mathcal{O} \subseteq \mathcal{M}$.
- At the beginning of the decoding, each message vertex v_i has an associated received value from the alphabet $\mathcal{O}$. The message vertex sends this value to each of its check neighbors.
- Each check vertex receives messages along all its incident edges. It processes the received set of values and sends back to each of its neighboring message vertices values from the alphabet $\mathcal{M}$.
- In a similar manner, each message vertex receives messages along all its incident edges, processes the received values and sends back to each of its neighbors values from the alphabet $\mathcal{M}$.
- The decoding process proceeds in iterations. In each iteration, values are transmitted iteratively back and forth between the two parts of the graph G.

1.3 Related work

In this section, we summarize the known results on LDPC codes. In Section 1.3.1, we consider works that analyze properties of LDPC codes decoded by message-passing algorithms. In

Section 1.3.2, we go over the constructions based on expander graphs. In Sections 1.3.3, we survey the results obtained using purely algebraic tools. Finally, some additional works are discussed in Section 1.3.4.

1.3.1 Message-passing algorithms

In Gallager's 1963 paper [34], there were two MP decoding algorithms presented for decoding of LDPC codes: one is a hard-decoding algorithm and the other is a soft-decoding one. An analysis of convergence of the algorithms was presented, and bounds on the decoding error probability were derived. The connection between the convergence of the algorithms and the non-existence of short cycles in the bipartite graph representing the code was considered.

The problem that limited Gallager's analysis in [34], was the existence of short cycles in the graph G. Gallager took into account in his analysis the fact that the shortest cycles in the graph G have length logarithmic in the number of message vertices. However, it was found by simulations that even graphs with short cycles produce LDPC codes that perform well in practice. While the analysis of Gallager took care of a specific graph, Urbanke and Richardson in [69] has overcome the problem of short cycles by taking into consideration a *random* bipartite graph. They showed that the average behavior of a message-passing decoding algorithm on a random bipartite graph converges to the cycle-free case.

The critical rate of LDPC codes, with respect to the MP decoding algorithm, was studied by Richardson and Urbanke in [69]. It was shown that for almost all LDPC codes in a suitably defined ensemble, transmissions at rates below the critical rate (that depends on the channel and is smaller than the Shannon capacity) can be obtained with MP error decoding probability that approaches zero exponentially fast in the length of the code. On the other hand, for transmissions at rates above the critical rate, the MP error decoding probability stays bounded away from zero. An effective algorithm for determining this rate for the binary symmetric channel with a message-passing decoding algorithm was proposed in [69].

In [70], very low-density parity-check codes were designed with rates extremely close to the Shannon capacity. In particular, they clearly outperform turbo codes. However, those codes were found by a computer search and not as part of a general construction.

1.3.2 Expander constructions

The notion of low-complexity codes was introduced by Tanner in [82]. In that paper, the author suggested a modified construction for LDPC codes, based on a bipartite graph with each message vertex connected to exactly two check vertices (see definition (1.2)). Tanner derived a lower bound on the rate and the minimum distance of the presented codes.

Encoding and decoding method for the proposed codes were also presented.

Expander graphs have attracted many investigators over the years. There are known explicit constructions of Ramanujan expander graphs due to Lubotsky, Philips, and Sarnak [22, Chapter 4], [53] and Margulis [62]. In particular, Lubotsky *et al.* have shown in [53] that for any prime $p_0 \equiv 1 \pmod 4$ there exists an infinite family of bipartite $(p_0 + 1)$-regular Ramanujan graphs with the number of vertices approaching infinity. Independently, Margulis presented in [62] similar result.

The explicit constructions of Ramanujan graphs were used by Alon *et al.* in [2] as building blocks to obtain new polynomial-time constructions of asymptotically good codes in the low-rate range.

Using expander graphs, a family of explicit expander-based LDPC codes was presented in [79] by Sipser and Spielman. These codes have linear-time sequential decoding (yet not encoding) algorithms and logarithmic-time parallel decoding algorithms that use a linear number of processors. The codes are asymptotically good, i.e. both the code dimension and the number of correctable errors grow linearly with the code length.

Another family of explicit low-density error-correcting codes was presented in [81] by Spielman. By combining ideas from [2] and [79], Spielman provided in [81] an asymptotically good construction where both the decoding and encoding time complexities were linear in the code length. However, these codes exist at rates lower than the rates of their counterparts in [79].

The expander graph approach was explored by Burshtein and Miller in [20] for LDPC codes. The authors considered codes represented by an expander graph. They proved that both the hard-decoding and the soft-decoding message-passing algorithms, when applied to these codes, can correct a number of errors that is linear in the code length. The implication of this result is that when the block length is sufficiently large, once a message-passing algorithm has corrected a sufficiently large fraction of the errors, it will eventually correct all errors. This result was combined with known results [69] on the ability of message-passing algorithms to reduce the number of errors to an arbitrarily small fraction for relatively high transmission rates. Consequently, the following two-step decoding strategy will correct all errors:

- In the first step, all but a small portion of the errors are corrected by the message-passing algorithms as proposed in [69] .
- In the second step, the rest of the errors are corrected by message-passing algorithms, following the result in [20].

The results of Burshtein and Miller hold for various message-passing algorithms, including Gallager's original hard-decision and soft-decision decoding algorithms.

Tanner's low-complexity codes were further studied by Etzion *et al.* in [25]. For low complexity codes with cycle-free graphs the authors showed decoding algorithm with time complexity quadratic in the code length. They also proved that cycle-free graphs produce rather poor codes.

While the linear-time decoder of the Sipser-Spielman construction was guaranteed to correct a number of errors that is a positive fraction of the code length, that fraction was significantly smaller than what one could attain by bounded-distance decoding—namely, decoding up to half the minimum distance of the code. The guaranteed fraction of linear-time correctable errors was substantially improved by Zémor in [84]. In his analysis, Zémor considered the special (yet abundant) case of the Sipser-Spielman construction where the underlying Ramanujan graph is bipartite, and presented a linear-time iterative decoder where the correctable fraction was 1/4 of the relative minimum distance of the code. We will describe the work of Zémor in more detail in Section 1.4.

In a series of works [8], [9], [10], Barg and Zémor generalize the construction from [84]. The results in these papers are essential for our discussion, and thus we will discuss them in more detail in Section 1.4.

Guruswami and Indyk in [36] and [37] improved the results of Zémor [84] and presented a linear-time encodable and decodable expander-based codes that can correct a number of errors equal to about half the minimum distance of their code. They presented codes of low rate $\mathcal{R}$ with relative minimum distance of about $1/2 - 2\sqrt[4]{\mathcal{R}}$.

In [38], Guruswami and Indyk used Zémor's construction as a building block and combined it with methods from [2], [4], and [5] to suggest a code construction with the following three properties:

(P1) The construction is nearly-MDS: it yields for every design rate $\mathcal{R} \in (0, 1]$ and sufficiently small $\epsilon > 0$ an infinite family of codes of rate at least $\mathcal{R}$ over an alphabet of size

$$2^{O\left((\log(1/\epsilon))/(\mathcal{R}\epsilon^4)\right)} , \qquad (1.6)$$

and the relative minimum distance of the codes is greater than

$$1 - \mathcal{R} - \epsilon .$$

(P2) The construction is linear-time encodable, and the time complexity per symbol is $\textsc{Poly}(1/\epsilon)$ (i.e., this complexity grows polynomially with $1/\epsilon$).

(P3) The construction has a linear-time decoder which is essentially a bounded-distance decoder: the correctable number of errors is at least a fraction $(1-\mathcal{R}-\epsilon)/2$ of the code length. The time complexity per symbol of the decoder is also $\textsc{Poly}(1/\epsilon)$.

In fact, the decoder described by Guruswami and Indyk in [38] is more general in that it can handle a combination of errors and erasures. Thus, by using their codes as an outer code in a concatenated construction, one obtains a linear-time encodable code that attains the Zyablov bound [23, p. 1949], with a linear-time bounded-distance decoder. Alternatively, such a concatenated construction approaches the capacity of any given memoryless channel: if the inner code is taken to have the smallest decoding error exponent, then the overall decoding error probability behaves like Forney's error exponent [30], [31] (the time complexity of searching for the inner code, in turn, depends on ϵ, yet not on the overall length of the concatenated code).

A family of LDPC codes constructed from Ramanujan graphs was considered by Rosenthal and Vontobel in [76]. The authors presented simulation results on the performance of the constructed codes.

1.3.3 Algebraic approach

Over the decades, algebraic code constructions were widely investigated by researchers. In several recent works, the algebraic approach was also applied to low-density parity-check codes. The reader can refer, for example, to [14], [16], [17], [64].

Algebraic MDS codes over $(\mathrm{GF}(2))^{k-1}$, k prime, were constructed by Blaum et *al.* [14]. Simple encoding and decoding algorithms were presented for these codes. The authors showed that when the symbols of $\mathrm{GF}(2^{k-1})$ in any codeword are concatenated to form a binary word, the produced binary code, which is linear over GF(2), has the LDPC property. Lower and upper bounds on the bit density of the parity-check matrix of the code were derived.

Another LDPC code construction over $(\mathrm{GF}(q))^k$, where k is a positive integer, was presented by Blaum and Roth in [17]. A bound on the lowest possible density of a parity-check matrix (over $\mathrm{GF}(q)$) of an MDS code (over $(\mathrm{GF}(q))^k$) was derived. Codes that achieve the mentioned bound were presented for certain redundancy values together with encoding and decoding algorithms.

Other algebraic constructions were presented, for example, in [27] and [64]. In [15], some algebraic LDPC codes were compared through simulation of the message-passing decoding algorithm.

1.3.4 Other results

Over the past years, LDPC codes were often constructed using regular bipartite graphs. It was shown by Luby *et al.* in [55] that at a given rate, codes based on irregular graphs may provide lower decoding error probability compared to codes based on regular graphs. The authors obtained some condition for testing whether any given degree sequence minimizes the decoding error probability under the MP decoding. This condition can be formulated in the form of a linear-programming problem. By numerically solving that linear-programming problem, the authors of [55] were able to select degree sequences that characterize codes with low decoding error probability. However, no constructive method for finding such graphs was given.

In [71], an *encoding* method for codes that are based on sparse matrices was presented. For randomly picked code graph, the encoding complexity of this method is linear in the code length n with probability approaching 1 as $n \to \infty$.

Kou, Lin and Fossorier proposed a construction of LDPC codes based on finite geometries [48]. Although the proposed construction was simple, it was designed for specific parameters and thus it was not good asymptotically. The proposed codes have a linear-time encoding algorithm. The performance of several codes from this family was studied by simulation and was found to be close to the Shannon capacity.

Litsyn and Shevelev studied in [51], [52] the distance distribution of several random ensembles of irregular and regular LDPC codes, respectively. For regular codes, the compared ensembles include an ensemble defined by random parity-check matrices having given column and row sum, ensembles defined by random matrices with given column sums or given row sums, and an ensemble defined by random bipartite graphs. For irregular codes, the considered ensemble is defined by a certain set of random parity-check matrices.

1.4 Introducing expander codes

1.4.1 Expander codes of Sipser and Spielman

Construction

The expander codes described herein were the first asymptotically-good codes that allowed a linear-time (in their length) decoding algorithm, which is able to correct a constant fraction of errors. The construction of these codes is due to Sipser and Spielman [79], and it can be described in terms of graphs as shown below.

Let $\mathcal{G} = (V, E)$ be a Δ-regular undirected graph with a vertex set V, $|V| = n$, and an edge set E of size $N = \frac{1}{2}n\Delta$. We assume an ordering on V, thereby inducing an ordering on the edges of $E(v)$ for every $v \in V$. Let $\mathbb{F} = \mathrm{GF}(2)$ and fix $\mathcal{C}$ to be a linear $[\Delta, k{=}r\Delta, d_0{=}\delta_0\Delta]$ code over $\mathbb{F}$. The code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ is defined in [79] to be a low-complexity code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ over $\mathbb{F}$ (with respect to the graph $\mathcal{G}$ and the code $\mathcal{C}$ as above), when the underline graph $\mathcal{G}$ is taken to be an expander graph. Then, $\mathbb{C}$ is a linear $[N, K, D]$ code. It is known that

$K/N \geq 2r-1$ [82], and it was shown in [79] that

$$D \geq N \left(\frac{\delta_0 - \gamma_{\mathcal{G}}}{1 - \gamma_{\mathcal{G}}} \right)^2 = N \left(\delta_0^2 - O(\gamma_{\mathcal{G}}) \right) \ , \tag{1.7}$$

where $\gamma_{\mathcal{G}}$ is the ratio between the second largest eigenvalue of the adjacency matrix of $\mathcal{G}$ and Δ. Thus, when the underlying graphs have a good separation between the first and the second eigenvalue, namely $\gamma_{\mathcal{G}} \to 0$ for $\Delta \to \infty$, the minimum distance of the codes $\mathbb{C}$ approaches $N\delta_0^2$. Such graphs can be taken from the families of Ramanujan graphs, for example those from [53] or [62].

When the codes $\mathcal{C}$ are taken to be random binary codes of rate r and relative minimum distance $\delta_{GV}(r) = \mathsf{H}_2^{-1}(1-r)$, it can be concluded from inequality (1.7) that the codes $\mathbb{C}$ of rate $\mathcal{R}$ have relative minimum distance

$$\delta \geq \left(\mathsf{H}_2^{-1} \left(\tfrac{1}{2}(1-\mathcal{R}) \right) \right)^2 \ .$$

The construction in [79] can be generalized to larger fields $\mathbb{F}$. When $\mathbb{F} \geq \Delta$, the constituent code $\mathcal{C}$ can be taken as a GRS code, thus resulting in a better relative minimum distance of the code $\mathbb{C}$ (over the alphabet $\mathbb{F}$), which becomes in that case

$$\delta \geq \left(\tfrac{1}{2}(1-\mathcal{R}) \right)^2 \ .$$

Decoding

In [79], the decoder for the *binary* code $\mathbb{C}$ is presented. The decoder has time complexity which is linear in the overall code length N. That decoder is able to correct number of errors which is a fraction

$$\frac{1}{48} \left(\mathsf{H}_2^{-1} \left(\tfrac{1}{2}(1-\mathcal{R}) \right) \right)^2$$

of the code lenght; namely, the fraction is only about $\frac{1}{48}$ of the code relative minimum distance. Still, the result has high importance since it was the first linear-time decoder that was able to correct a constant fraction of errors (which was not dependent on the length of the code). Here, we omit the details of the decoder.

1.4.2 Expander codes of Zémor

Construction

In [84], Zémor considered a special class of the Sipser-Spielman construction of expander codes, where the underlying graph is bipartite [79], [82]. We summarize the construction next.

Let $\mathcal{G} = (V, E)$ be a Δ-regular bipartite undirected graph with a vertex set $V = A \cup B$ such that $A \cap B = \emptyset$ and $|A| = |B| = n$, and an edge set E of size $N = n\Delta$ such that every edge in E has one endpoint in A and one endpoint in B. We assume an ordering on V, thereby inducing an ordering on the edges of $E(v)$ for every $v \in V$.

Let $\mathbb{F} = \mathrm{GF}(q)$. Observe that the subset A (respectively, B) induces on every word $\boldsymbol{x} \in \mathbb{F}^N$ a partition into n non-overlapping sub-words $(\boldsymbol{x})_{E(v)} \in \mathbb{F}^\Delta$, where v ranges over the elements of A (respectively, B).

Fix $\mathcal{C}$ to be a linear $[\Delta, k{=}r\Delta, d_0{=}\delta_0\Delta]$ code over $\mathbb{F}$. The code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ is defined in [84] to be a low-complexity code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ over $\mathbb{F}$ (with respect to the graph $\mathcal{G}$ and the code $\mathcal{C}$ as above). The $\mathbb{C}$ is a linear $[N, K, D]$ code. It also holds in this case that $K/N \geq 2r{-}1$, and that

$$D \geq N \left(\frac{\delta_0 - \gamma_{\mathcal{G}}}{1 - \gamma_{\mathcal{G}}} \right)^2 = N \left(\delta_0^2 - O(\gamma_{\mathcal{G}}) \right) \ , \tag{1.8}$$

where $\gamma_{\mathcal{G}}$ is the ratio between the second largest eigenvalue of the adjacency matrix of $\mathcal{G}$ and Δ. If the graph $\mathcal{G}$ in the construction of $\mathbb{C}$ is taken to be a Ramanujan graph, we obtain from (1.8) that the $O(\cdot)$ expression goes to zero as Δ becomes large.

Decoding

The iterative decoding algorithm of Zémor is shown in Figure 1.3, where $\mathcal{D} : \mathbb{F}^\Delta \to \mathcal{C}$ stands for a decoder for $\mathcal{C}$ that recovers correctly any pattern of less than $d_0/2$ errors. It is shown in [84] that the algorithm in Figure 1.3 can correct any error word whose Hamming weight does not exceed

$$\frac{1}{2} \cdot \alpha N \delta_0 \left(\frac{\delta_0}{2} - \gamma_{\mathcal{G}} \right) = \frac{1}{4} \cdot \alpha N \left(\delta_0^2 - O(\gamma_{\mathcal{G}}) \right) \ ,$$

for any fixed positive constant $\alpha < 1$. The number of iterations ν in Figure 1.3 can be taken as $\lfloor (\log n)/\log(2{-}\alpha) \rfloor$. Using similar arguments as in [79], it can be shown that Zémor's algorithm can be implemented in time complexity $O(N)$, assuming that α and the code $\mathcal{C}$ are fixed (in particular, this assumption implies that $\mathcal{D}$ can be implemented in constant time).

1.4.3 Expander codes of Barg and Zémor (2002)

Construction

We briefly recall here the construction and the decoder in [10]. Let $\mathcal{G} = (V, E)$ be a bipartite Δ-regular undirected connected graph with a vertex set $V = A \cup B$, as in Section 1.4.2.

Let $\mathcal{C}_A$ and $\mathcal{C}_B$ be two linear codes of length Δ over $\mathbb{F}$ (this $\mathbb{F}$ will be defined below). The

```
Input: Received word $\boldsymbol{y} = (y_e)_{e \in E}$.
Let $\boldsymbol{z} \leftarrow \boldsymbol{y}$.
For $i \leftarrow 1$ to $\nu$ do {
    Let $X$ stand for $A$ if $i$ is odd, and for $B$ otherwise.
    Iteration $i$: For every $v \in X$ let $(\boldsymbol{z})_{E(v)} \leftarrow \mathcal{D}\left((\boldsymbol{z})_v\right)$.
}
Output: $\boldsymbol{z}$.
```

Figure 1.3: Zémor's decoder in [84].

code $\mathbb{C}_{BZ2} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ is defined as

$$\begin{aligned} \mathbb{C}_{BZ2} \;\; = \;\; & \big\{\boldsymbol{c} \in \mathbb{F}^N \, : \, (\boldsymbol{c})_{E(u)} \in \mathcal{C}_A \text{ for every } u \in A \\ & \qquad \text{and } \; (\boldsymbol{c})_{E(u)} \in \mathcal{C}_B \text{ for every } \; u \in B\big\} \; , \end{aligned} \tag{1.9}$$

where $(\boldsymbol{x})_{E(u)}$ denotes the sub-word of $\boldsymbol{x} = (x_e)_{e \in E} \in \mathbb{F}^N$ that is indexed by $E(u)$. The produced code $\mathbb{C}$ is a linear code of length N over $\mathbb{F}$.

The authors of [10] consider two separate cases.

Case 1: $\mathbb{F} = \mathrm{GF}(2)$. In this case, the codes $\mathcal{C}_A$ and $\mathcal{C}_B$ are chosen to have the best possible decoding error probability under maximum-likelihood decoding.

Case 2: $\mathbb{F} = (\mathrm{GF}(2))^\ell$, for $\ell \in \mathbb{N}$. Fix some small $\epsilon > 0$. The code $\mathcal{C}_A$ is taken as a linear $[\Delta\ell, r_A\Delta\ell, \delta_A\Delta\ell]$ binary code. The code $\mathcal{C}_A$ can also be thought as a linear $[\Delta, r_A\Delta, D_A]$ code over $\mathbb{F}$. In addition, $\mathcal{C}_A$ is chosen to satisfy all the following properties:

(a) $\mathcal{C}_A$ has the best possible error probability under maximum-likelihood decoding;
(b) $\delta_A \geq \mathsf{H}_2^{-1}(1 - r_A) - \epsilon$;
(c) $D_A \geq (1 - r_A)\Delta - \epsilon$.

The code $\mathcal{C}_B$ is defined similarly to (a)-(c) with respect to its parameters.

Decoding

Let us transmit a codeword $\boldsymbol{c} = (c_e)_{e \in E} \in \mathbb{C}_{BZ2}$ through a BSC with crossover probability p. Assume that $\boldsymbol{y} = (y_e)_{e \in E}$ is the received (erroneous) word. A formal definition of the decoder $\mathcal{D}_{BZ2}$ appears in Figure 1.4. The number of iterations ν is taken to be $O(\log n)$.

```
Input:  Received word $\boldsymbol{y} = (y_e)_{e \in E}$ in $\mathbb{F}^N$.
Let $\boldsymbol{z} \leftarrow \boldsymbol{y}$.
For $i \leftarrow 1, 2, \ldots, \nu$ do {
    If $i$ is odd  then $X \equiv A$, $\mathcal{D} \equiv \mathcal{D}_A$,
        else $X \equiv B$, $\mathcal{D} \equiv \mathcal{D}_B$.
    For $u \in X$ do $(\boldsymbol{z})_{E(u)} \leftarrow \mathcal{D}((\boldsymbol{z})_{E(u)})$.
}
Output:  $\boldsymbol{z}$ if $\boldsymbol{z} \in \mathbb{C}_{BZ2}$ (and declare 'error' otherwise).
```

Figure 1.4: Decoder $\mathcal{D}_{BZ2}$ of Barg and Zémor for the code $\mathbb{C}_{BZ2}$.

The decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ are the *maximum-likelihood* decoders for the codes $\mathcal{C}_A$ and $\mathcal{C}_B$, respectively.

Results

Fix a design code rate $\mathcal{R} < \mathsf{C}_2(p)$. In [10], Barg and Zémor show that for $\mathbb{F} = \mathrm{GF}(2)$, the decoding error probability of the code $\mathbb{C}_{BZ2}$, under the decoder in Figure 1.4, is bounded by

$$\mathsf{Prob}_e(\mathbb{C}_{BZ2}) \leq \exp\{-\alpha N f_3(\mathcal{R})\} \ ,$$

where $\alpha \in (0,1)$ is a constant, and $f_3(\mathcal{R})$ is bounded by

$$f_3(\mathcal{R}) \geq \max_{\mathcal{R} \leq r < \mathsf{C}_2(p)} \left\{ \mathbb{E}(r) \cdot \left(\tfrac{1}{2} \mathsf{H}_2^{-1}(r - \mathcal{R}) - \Theta\left(\frac{1}{\sqrt{\Delta}} \right) \right) \right\} \ .$$

Moreover, it is shown in [10] that for a code $\mathbb{C}_{BZ2}$ over $(\mathrm{GF}(2))^\ell$ with the constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$ satisfying properties (a)-(c) above, and for the decoder in Figure 1.4, the decoding error probability is bounded by

$$\mathsf{Prob}_e(\mathbb{C}_{BZ2}) \leq \exp\{-\alpha N f_2(\mathcal{R})\} \ ,$$

where $\alpha \in (0,1)$ is a constant, and $f_2(\mathcal{R})$ is bounded by

$$f_2(\mathcal{R}) \geq \max_{\mathcal{R} \leq r < \mathsf{C}} \left\{ \mathbb{E}(r) \cdot \left(\tfrac{1}{2}(r - \mathcal{R}) - \Theta\left(\frac{1}{\sqrt{\Delta}} \right) \right) \right\} \ .$$

1.4.4 Expander codes of Barg and Zémor (2003)

Construction

We recall here the construction of the expander codes presented in [8]. Let $\mathcal{G} = (V, E)$ be a bipartite graph with $V = V_0 \cup (V_1 \cup V_2)$, such that each edge has one endpoint in V_0 and one endpoint in either V_1 or V_2. Let $|V_i| = n$ for $i = 0, 1, 2$. Let the degree of each vertex in V_0, V_1, and V_2 be Δ, Δ_1, and $\Delta_2 = \Delta - \Delta_1$, respectively. In addition, let the subgraph $\mathcal{G}_1$ induced by $V_0 \cup V_1$ be a regular bipartite Ramanujan graph and denote by E_1 its edge set. Let λ_1 be a second largest eigenvalue of the adjacency matrix of $\mathcal{G}_1$.

Take $\ell \in \mathbb{N}$. Let $\mathcal{C}_A$ be a linear $[l\Delta, R_0 l\Delta, d_0 = l\Delta\delta_0]$ binary code of rate $R_0 = \Delta_1/\Delta$. It can also be thought as a linear code of length Δ over $\mathbb{F} = (\mathrm{GF}(2))^{\ell}$. Let $\mathcal{C}_B$ be a linear $[\Delta_1, R_1\Delta_1, d_1 = \Delta_1\delta_1]$ code over $\mathbb{F}$, and let $\mathcal{C}_{aux}$ be a code of length Δ_1 over $\mathbb{F}$. The code $\mathbb{C}_{BZ3}$ is defined as the set of vectors $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)$, indexed by the set E of size $N = \Delta n$, such that

1. For every vertex $v \in V_0$, the subvector $(\boldsymbol{x})_{E(v)}$ is a codeword of $\mathcal{C}_A$ (over $\mathbb{F}$) and the set of coordinates $E_1(v)$ is an information set for the code $\mathcal{C}_A$.
2. For every vertex $v \in V_1$, the subvector $(\boldsymbol{x})_{E(v)}$ is a codeword of $\mathcal{C}_B$ (over $\mathbb{F}$).
3. For every vertex $v \in V_0$, the subvector $(\boldsymbol{x})_{E_1(v)}$ is a codeword of $\mathcal{C}_{aux}$ (over $\mathbb{F}$).

Decoding

The authors of [8] proposed a decoding algorithm for the code $\mathbb{C}_{BZ3}$. In the first iteration, each subvector $\boldsymbol{z}(v)$, $v \in V_0$, is treated as follows: the decoder computes, for every symbol b of the q-ary alphabet, and for every edge $e \in E_1$ incident to v, the weight of the edge as follows:

$$d_{e,b}(\boldsymbol{z}) = \min_{\boldsymbol{a} \in \mathcal{C}_A : a_e = b} \mathsf{d}_2(\boldsymbol{a}, (\boldsymbol{z})_{E(v)}),$$

where a_e denotes the q-ary coordinate of the codeword $\boldsymbol{a}$ that corresponds to the edge e, and $\mathsf{d}_2(\cdot,\cdot)$ is the *binary* Hamming distance. This information is passed along the edge e to the corresponding decoder on the right-hand side of the bipartite graph. In the second iteration, for every vertex $w \in V_1$ the decoder associated to it finds a q-ary codeword $\boldsymbol{b} = (b_1, \ldots, b_{\Delta_1}) \in \mathcal{C}_B$ that satisfies

$$\boldsymbol{b} = \arg \min_{\boldsymbol{x} = (x_1, \ldots, x_{\Delta_1}) \in \mathcal{C}_B} \sum_{i=1}^{\Delta_1} d_{w(i), x_i}(\boldsymbol{z}) \ ,$$

and writes b_i on the edge $w(i)$, $i = 1, \ldots, \Delta_1$.

Then, the decoder continues similarly to the decoder in [10].

Results

It is shown in [8] that the decoding error probability of the code $\mathbb{C}_{BZ3}$, $\mathsf{Prob}_e(\mathbb{C}_{BZ3})$, satisfies

$$\mathsf{Prob}_e(\mathbb{C}_{BZ3}) \leq \exp\Big\{ -n\Delta l \delta_1 (1+\alpha)^{-1} \cdot (\mathbb{E}(R_0) - M\alpha)(1-o(1))\Big\},$$

where α is a constant defined in [8] (in paritcular, $1 > \alpha > 2\lambda_1/d_1$), and

$$M = M(R,p) = \begin{cases} \frac{1}{2}\log_2((1-p)/p) & \text{if } R \leq R_{crit} \\ \log_2\left(\frac{\delta_{GV}(R)(1-p)}{(1-\delta_{GV}(R))p}\right) & \text{if } R \geq R_{crit} \end{cases},$$

$\delta_{GV}(R) = \mathsf{H}_2^{-1}(1-R)$ is the Gilbert-Varshamov relative distance for the rate R, and $R_{crit} = 1 - \mathsf{H}_2(\rho_0)$ is a so-called *critical rate*, where $\rho_0 = \sqrt{p}/(\sqrt{p} + \sqrt{1-p})$ (see [8] for details). This decoding error probability could be made arbitrarily close to the decoding error probability of concatenated codes (1.3) by taking significantly large codes.

It is also shown in [8] that the minimum distance of the code $\mathbb{C}_{BZ3}$ over GF(2) is bounded from below by

$$\delta_0\delta_1\left(1 - \frac{\lambda_1}{d_{aux}}\right)\left(1 - \frac{\lambda_1}{2d_1}\right) N \, , \tag{1.10}$$

and thus the codes $\mathbb{C}_{BZ3}$ approach the Zyablov bound (1.5) for significantly large values of N.

The decoder for the $\mathbb{C}_{BZ3}$, presented above, is capable of correcting a number of errors which is almost half of the lower bound in (1.10). This decoder has decoding time complexity which grows linearly with the code length.

1.4.5 Barg and Zémor's analysis of expander codes (2004)

In [9], using a more sophisticated analysis, the authors improve on the minimum-distance bounds for the codes described in Section 1.4.3 and Section 1.4.4.

In particular, for the codes $\mathbb{C}_{BZ2}$ of rate $\mathcal{R}$, they bounded the relative minimum distance from below by

$$\delta(\mathcal{R}) \geq \frac{1}{4}(1-\mathcal{R})^2 \cdot \min_{\delta_{GV}((1+\mathcal{R})/2) < \mathsf{B} < \frac{1}{2}} \frac{g(\mathsf{B})}{\mathsf{H}_2(\mathsf{B})} \, ,$$

where the function $g(\mathsf{B})$ is defined on page 28. For the codes $\mathbb{C}_{BZ3}$ of rate $\mathcal{R}$, the relative minimum distance is bounded from below by

$$\delta(\mathcal{R}) \geq \max_{\mathcal{R} \leq r \leq 1} \left\{ \min_{\delta_{GV}(r) < \mathsf{B} < \frac{1}{2}} \left(\delta_0(\mathsf{B}, r) \cdot \frac{1 - \mathcal{R}/r}{\mathsf{H}_2(\mathsf{B})} \right) \right\} \, ,$$

where the function $\delta_0(\mathsf{B}, r)$ is defined on page 28. In particular, it follows that the relative minimum distance of these two families of codes is higher than the Zyablov bound (1.5) for a wide range of code rates.

1.5 Summary

In Table 1.1, we compare several relevant research works. The notations $\mathcal{R}$ and N stand for the rate and overall length of the cited constructions. The results in the table are compared according to the following three parameters (if available):

- Error probability (as a function of N).
- Relative minimum distance (as a function of $\mathcal{R}$).
- Encoding/decoding complexity per entry of codeword.

The results appear in chronological order.

Work	Upper bound on error probability	Lower bound on relative minimum distance	Encoding/decoding complexity per bit	Remarks
Gallager [33], [34] (1962 – 1963)	${\mathsf{Z}_G}^{-\sqrt{N}}$, Z_G is some constant.	Asymptotically good codes.	N/A	Shannon capacity is not attained.
Forney [30], [31] (1966)	$\max\limits_{\mathcal{R}\leq r\leq 1-\mathsf{H}_q(p)} e^{-N\mathbb{E}(r)(1-\frac{\mathcal{R}}{r})}$	N/A	$O(N)$	Concatenated codes, nonconstructive.
Justesen [41] (1972)	N/A	$\max\limits_{\max\{\mathcal{R},1/2\}\leq r\leq 1}(1-\frac{\mathcal{R}}{r})\mathsf{H}_2^{-1}(1-r)$, attains the Zyablov bound.	$O(N)$	
Sipser, Spielman [79], [81] (1996)	N/A	$\frac{1}{48}(\mathsf{H}_2^{-1}(\frac{1}{2}(1-\mathcal{R})))^2$	$O(1)$ for decoding, $O(1)$ for encoding in [81].	Low rates in [81].
Richardson, Urbanke [69], [70], [71] (2001)	${\mathsf{Z}_{RU}}^{-N}$ Z_{RU} is some constant.	N/A	Decoding is $O(1)$, encoding is $O(1)$ with probability $\to 1$ when $N\to\infty$.	Random coding result. Degree sequences found by computer search, very close to the Shannon capacity.

Work	Upper bound on error probability	Lower bound on relative minimum distance	Encoding/decoding complexity per bit	Remarks
Barg, Zémor [10], [84] (2001 – 2002)	$\max\limits_{\mathcal{R}\le r\le 1-\mathsf{H}_q(p)} e^{-N\mathbb{E}(r)\frac{r}{2}(1-\frac{\mathcal{R}}{r})}$	$\frac{1}{2}(1-\mathcal{R})\mathsf{H}_2^{-1}(\frac{1}{2}(1-\mathcal{R}))$	$O(1)$ Decoding only.	
Guruswami, Indyk [36], [37], [38] (2002)	N/A	$\max\limits_{\mathcal{R}\le r\le 1}(1-\frac{\mathcal{R}}{r})\mathsf{H}_2^{-1}(1-r)$, attains the Zyablov bound.	$O(1)$	
Barg, Zémor [8], [9] (2003 – 2004)	$\max\limits_{\mathcal{R}\le r\le 1-\mathsf{H}_q(p)} e^{-N\mathbb{E}(r)(1-\frac{\mathcal{R}}{r})}$ attains Forney's exponent.	$\frac{1}{4}(1-\mathcal{R})^2\cdot\min\limits_{\delta_{GV}((1+\mathcal{R})/2)<\mathsf{B}<1/2}(g(\mathsf{B})/\mathsf{H}_2(\mathsf{B}))$ for high rates. $\max\limits_{\mathcal{R}\le r\le 1}\left\{\min\limits_{\delta_{GV}(r)<\mathsf{B}<\frac{1}{2}}\left(\delta_0(\mathsf{B},r)\cdot\frac{1-\mathcal{R}/r}{\mathsf{H}_2(\mathsf{B})}\right)\right\}$ for low rates. Surpasses the Zyablov bound.	$O(1)$ Decoding only.	$g(\mathsf{B})$ and $\delta_0(\mathsf{B},r)$ are defined on page 28.

Table 1.1: Comparison of known results

Definitions of the functions $g(\mathsf{B})$ and $\delta_0(\mathsf{B}, r)$

Let B_1 be the largest root of the equation

$$\mathsf{H}_2(\mathsf{B}) = \mathsf{H}_2(\mathsf{B}) \left(\mathsf{B} - \mathsf{H}_2(\mathsf{B}) \cdot \frac{\delta_{GV}(\mathcal{R})}{1-\mathcal{R}} \right) = -\left(\mathsf{B} - \delta_{GV}(\mathcal{R}) \right) \cdot \log_2(1-\mathsf{B}) \; .$$

Moreover, let

$$a_1 = \frac{\mathsf{B}_1}{\mathsf{H}_2(\mathsf{B}_1)} - \frac{\delta_{GV}(\mathcal{R})}{\mathsf{H}_2(\delta_{GV}(\mathcal{R}))} \; ,$$

and

$$b_1 = \frac{\delta_{GV}(\mathcal{R})}{\mathsf{H}_2(\delta_{GV}(\mathcal{R}))} \cdot \mathsf{B}_1 - \frac{\mathsf{B}_1}{\mathsf{H}_2(\mathsf{B}_1)} \cdot \delta_{GV}(\mathcal{R})) \; .$$

The function $g(\mathsf{B})$ is defined in [9] as

$$g(\mathsf{B}) = \begin{cases} \dfrac{\delta_{GV}(\mathcal{R})}{1-\mathcal{R}} & \text{if } \mathsf{B} \leq \delta_{GV}(\mathcal{R}) \\ \dfrac{\mathsf{B}}{\mathsf{H}_2(\mathsf{B})} & \text{if } \delta_{GV}(\mathcal{R}) \leq \mathsf{B} \text{ and } \mathcal{R} \leq 0.284 \\ \dfrac{a_1\mathsf{B} + b_1}{\mathsf{B}_1 - \delta_{GV}(\mathcal{R})} & \text{if } \delta_{GV}(\mathcal{R}) \leq \mathsf{B} \leq \mathsf{B}_1 \text{ and } 0.284 < \mathcal{R} \leq 1 \\ \dfrac{\mathsf{B}}{\mathsf{H}_2(\mathsf{B})} & \text{if } \mathsf{B}_1 < \mathsf{B}_1 \leq 1 \text{ and } 0.284 < \mathcal{R} \leq 1 \end{cases} .$$

The function $\delta_0(\mathsf{B}, r)$ is defined to be $\omega^{\star\star}(\mathsf{B})$ for $\delta_{GV}(r) \leq \mathsf{B} \leq \mathsf{B}_1$, where

$$\omega^{\star\star}(\mathsf{B}) = r\mathsf{B} + (1-r)\mathsf{H}_2^{-1}\left(1 - \frac{r}{1-r}\mathsf{H}_2(\mathsf{B})\right) \; ,$$

and B_1 is the only root of the equation

$$\delta_{GV}(r) = w^{\star}(\mathsf{B}) \; ,$$

where

$$w^{\star}(\mathsf{B}) = (1-r)\left((2^{\mathsf{H}_2(\mathsf{B})/\mathsf{B}} + 1)^{-1} + \frac{\mathsf{B}}{\mathsf{H}_2(\mathsf{B})}\left(1 - \mathsf{H}_2\left((2^{\mathsf{H}_2(\mathsf{B})/\mathsf{B}} + 1)^{-1}\right)\right) \right) \; .$$

For $\mathsf{B}_1 \leq \mathsf{B} \leq \frac{1}{2}$, the function $\delta_0(\mathsf{B}, r)$ is defined to be a tangent to the function $\omega^{\star\star}(\mathsf{B})$ drawn from the point $\left(\frac{1}{2}, \omega^{\star}(\frac{1}{2})\right)$.

Chapter 2

Nearly-MDS expander codes

The material in this section has been published as [74], [75].

In this chapter, we present a family of codes which improves on the Guruswami-Indyk construction [38], which was mentioned in Section 1.3.2. Specifically, our codes will satisfy properties (P1)–(P3), except that the alphabet size in property (P1) will now be only

$$2^{O\left((\log(1/\epsilon))/\epsilon^3\right)} . \tag{2.1}$$

The basic ingredients of our construction are similar to those used in [38] (and also in [4] and [5]), yet their layout (in particular, the order of application of the various building blocks), and the choice of parameters will be different. Our presentation will be split into two parts. We first describe in Section 2.1 a construction that satisfies only the two properties (P1) and (P3) over an alphabet of size (2.1). These two properties will be proved in Sections 2.2 and 2.3. We also show that the codes studied by Barg and Zémor in [10] and [8] can be seen as concatenated codes, with our codes serving as the outer codes.

The second part of our presentation consists of Section 2.4, where we modify the construction of Section 2.1 and use the resulting code as a building block in a second construction, which satisfies property (P2) as well.

2.1 Construction of linear-time decodable codes

Let $\mathcal{G} = (V = A \cup B,\ E)$ be a bipartite Δ-regular graph defined in Section 1.4.2, with $|A| = |B| = n$, $|E| = n\Delta$. We restrict ourselves to simple graphs $\mathcal{G}$, i.e. graphs with no parallel edges and no self-loops. We will assume hereafter without any practical loss of generality that $n > 1$. As before, we use the notation $E(u)$ for the set of edges that are incident with u, and $(\boldsymbol{z})_{E(u)}$ for the sub-block of $\boldsymbol{z}$ that is indexed by $E(u)$ (we assume an ordering on V, thereby inducing an ordering on the edges of $E(u)$ for every $u \in V$).

Let $\mathbb{F}$ be the field $\mathrm{GF}(q)$ and let $\mathcal{C}_A$ and $\mathcal{C}_B$ be linear $[\Delta, r_A\Delta, \delta_A\Delta]$ and $[\Delta, r_B\Delta, \delta_B\Delta]$ codes over $\mathbb{F}$, respectively. We define the code $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ similarly to the definition of $\mathbb{C}_{BZ2}$ in (1.9), namely, as the linear code of length $|E|$ over $\mathbb{F}$:

$$\mathbb{C} = \Big\{ \boldsymbol{c} \in \mathbb{F}^{|E|} \,:\, (\boldsymbol{c})_{E(u)} \in \mathcal{C}_A \text{ for every } u \in A \\ \text{and } (\boldsymbol{c})_{E(v)} \in \mathcal{C}_B \text{ for every } v \in B \Big\} \;. \tag{2.2}$$

Let Φ be the alphabet $\mathbb{F}^{r_A\Delta}$. Fix some linear one-to-one mapping $\mathcal{E} : \Phi \to \mathcal{C}_A$ over $\mathbb{F}$, and let the mapping $\psi_{\mathcal{E}} : \mathbb{C} \to \Phi^n$ be given by

$$\psi_{\mathcal{E}}(\boldsymbol{c}) = \big(\mathcal{E}^{-1}\left((\boldsymbol{c})_{E(u)}\right)\big)_{u \in A} \;, \quad \boldsymbol{c} \in \mathbb{C} \;. \tag{2.3}$$

That is, the entries of $\psi_{\mathcal{E}}(\boldsymbol{c})$ are indexed by A, and the entry that is indexed by $u \in A$ equals $\mathcal{E}^{-1}\left((\boldsymbol{c})_{E(u)}\right)$. We now define the code $\mathbb{C}_\Phi$ of length n over Φ by

$$\mathbb{C}_\Phi = \{\psi_{\mathcal{E}}(\boldsymbol{c}) \,:\, \boldsymbol{c} \in \mathbb{C}\} \;. \tag{2.4}$$

Every codeword $\boldsymbol{x} = (\boldsymbol{x}_u)_{u \in A}$ of $\mathbb{C}_\Phi$ (with entries $\boldsymbol{x}_u$ in Φ) is associated with a unique codeword $\boldsymbol{c} \in \mathbb{C}$ such that

$$\mathcal{E}(\boldsymbol{x}_u) = (\boldsymbol{c})_{E(u)} \;, \quad u \in A \;.$$

Based on the definition of $\mathbb{C}_\Phi$, the code $\mathbb{C}$ can be represented as a concatenated code with an inner code $\mathcal{C}_A$ over $\mathbb{F}$ and an outer code $\mathbb{C}_\Phi$ over Φ. It is possible, however, to use $\mathbb{C}_\Phi$ as an outer code with inner codes other than $\mathcal{C}_A$. Along these lines, the codes studied in [10] and [8] can be represented as concatenated codes with $\mathbb{C}_\Phi$ as an outer code, whereas the inner codes are taken over a sub-field of $\mathbb{F}$.

2.2 Bounds on the code parameters

Let $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$, Φ, and $\mathbb{C}_\Phi$ be as defined in Section 2.1. It was shown in [10] that the rate of $\mathbb{C}$ is at least

$$r_A + r_B - 1 \;. \tag{2.5}$$

From the fact that $\mathbb{C}$ is a concatenated code with an inner code $\mathcal{C}_A$ and an outer code $\mathbb{C}_\Phi$, it follows that the rate of $\mathbb{C}_\Phi$ is bounded from below by

$$\frac{r_A + r_B - 1}{r_A} = 1 - \frac{1}{r_A} + \frac{r_B}{r_A} \;. \tag{2.6}$$

In particular, the rate approaches r_B when $r_A \to 1$.

We next turn to computing a lower bound on the relative minimum distance of $\mathbb{C}_\Phi$. By applying this lower bound, we will then verify that $\mathbb{C}_\Phi$ satisfies property (P1). Our analysis is based on that in [8], and we obtain here an improvement over a bound that can be inferred from [8]; we will need that improvement to get the reduction of the alphabet size from (1.6) to (2.1). We first introduce several notations.

Denote by $A_\mathcal{G}$ the adjacency matrix of $\mathcal{G}$. It is known that Δ is the largest eigenvalue of $A_\mathcal{G}$. Let $\gamma_\mathcal{G}$ be the ratio between the second largest eigenvalue of $A_\mathcal{G}$ and Δ.

When $\mathcal{G}$ is taken from a sequence of Ramanujan expander graphs with constant degree Δ, such as the LPS graphs in [53], we have

$$\gamma_\mathcal{G} \le \frac{2\sqrt{\Delta-1}}{\Delta} \; .$$

For a nonempty subset S of the vertex set V of $\mathcal{G}$, we will use the notation $\mathcal{G}_S$ to stand for the subgraph of $\mathcal{G}$ that is induced by S: the vertex set of $\mathcal{G}_S$ is given by S, and its edge set, denoted by E_S, consists of all the edges in $\mathcal{G}$ that have each of their endpoints in S. The degree of u in $\mathcal{G}_S$, which is the number of adjacent vertices to u in $\mathcal{G}_S$, will be denoted by $\deg_S(u)$.

Theorem 2.2.1 *The relative minimum distance of the code $\mathbb{C}_\Phi$ is bounded from below by*

$$\frac{\delta_B - \gamma_\mathcal{G}\sqrt{\delta_B/\delta_A}}{1-\gamma_\mathcal{G}} \; .$$

In particular, this lower bound approaches δ_B when $\gamma_\mathcal{G} \to 0$.

The proof of the theorem will make use of Proposition 2.2.3 below, which is an improvement on Corollary 9.2.5 in Alon and Spencer [3] for bipartite graphs, and is also an improvement on Lemma 4 in Zémor [84]. We will need the following technical lemma for that proposition (which is a generalization of to the well known expander mixing lemma . The proof of this lemma can be found in Appendix A.

Lemma 2.2.2 *Let χ be a real function on the vertices of $\mathcal{G}$ where the images of χ are restricted to the interval $[0,1]$. Write*

$$\sigma = \frac{1}{n}\sum_{u\in A}\chi(u) \qquad \textit{and} \qquad \tau = \frac{1}{n}\sum_{v\in B}\chi(v) \; .$$

Then

$$\begin{aligned}\frac{1}{\Delta n}\sum_{u\in A}\sum_{v\in\mathcal{N}(u)}\chi(u)\chi(v) &\le \sigma\tau + \gamma_\mathcal{G}\sqrt{\sigma(1-\sigma)\tau(1-\tau)} \\ &\le (1-\gamma_\mathcal{G})\sigma\tau + \gamma_\mathcal{G}\sqrt{\sigma\tau} \; .\end{aligned}$$

(Comparing to the results in [84], Lemma 4 therein is stated for the special case where the images of χ are either 0 or 1. Our first inequality in Lemma 2.2.2 yields a bound which is always at least as tight as Lemma 4 in [84].)

Proposition 2.2.3 *Let $S \subseteq A$ and $T \subseteq B$ be subsets of sizes $|S| = \sigma n$ and $|T| = \tau n$, respectively, such that $\sigma + \tau > 0$. Then the sum of the degrees in the graph $\mathcal{G}_{S \cup T}$ is bounded from above by*

$$\sum_{u \in S \cup T} \deg_{S \cup T}(u) \le 2 \left((1-\gamma_{\mathcal{G}})\sigma\tau + \gamma_{\mathcal{G}}\sqrt{\sigma\tau} \right) \Delta n \ .$$

Proof. We select $\chi(u)$ in Lemma 2.2.2 to be

$$\chi(u) = \begin{cases} 1 & \text{if } u \in S \cup T \\ 0 & \text{otherwise} \end{cases} \ .$$

On the one hand, by Lemma 2.2.2,

$$\sum_{u \in A} \sum_{v \in \mathcal{N}(u)} \chi(u)\chi(v) \le \left((1-\gamma_{\mathcal{G}})\sigma\tau + \gamma_{\mathcal{G}}\sqrt{\sigma\tau} \right) \Delta n \ .$$

On the other hand,

$$2 \sum_{u \in A} \sum_{v \in \mathcal{N}(u)} \chi(u)\chi(v) = \sum_{u \in S \cup T} \deg_{S \cup T}(u) \ .$$

These two equations yield the desired result. □

Proof of Theorem 2.2.1. First, it is easy to see that $\mathbb{C}_\Phi$ is a linear subspace over $\mathbb{F}$ and, as such, it is an Abelian subgroup of Φ^n. Thus, the minimum distance of $\mathbb{C}_\Phi$ equals the minimum weight (over Φ) of any non-zero codeword of $\mathbb{C}_\Phi$.

Pick any non-zero codeword $\boldsymbol{x} \in \mathbb{C}_\Phi$, and let $\boldsymbol{c} = (c_e)_{e \in E}$ be the unique codeword in $\mathbb{C}$ such that $\boldsymbol{x} = \psi_{\mathcal{E}}(\boldsymbol{c})$. Denote by $Y \subseteq E$ the support of $\boldsymbol{c}$ (over $\mathbb{F}$), i.e.,

$$Y = \{ e \in E \ : \ c_e \ne 0 \} \ .$$

Let S (respectively, T) be the set of all vertices in A (respectively, B) that are endpoints of edges in Y. In particular, S is the support of the codeword $\boldsymbol{x}$. Let σ and τ denote the ratios $|S|/n$ and $|T|/n$, respectively, and consider the subgraph $\mathcal{G}(Y) = (S : T, Y)$ of $\mathcal{G}$. Since the minimum distance of $\mathcal{C}_A$ is $\delta_A \Delta$, the degree in $\mathcal{G}(Y)$ of every vertex in A is at least $\delta_A \Delta$. Therefore, the number of edges in $\mathcal{G}(Y)$ satisfies

$$|Y| \ge \delta_A \Delta \cdot \sigma n \ .$$

Similarly, the degree in $\mathcal{G}(Y)$ of every vertex in B is at least $\delta_B\Delta$ and, thus,

$$|Y| \geq \delta_B\Delta \cdot \tau n \ .$$

Therefore,

$$|Y| \geq \max\{\delta_A\sigma, \delta_B\tau\} \cdot \Delta n \ .$$

On the other hand, $\mathcal{G}(Y)$ is a subgraph of $\mathcal{G}_{S\cup T}$; hence, by Proposition 2.2.3,

$$|Y| \leq \frac{1}{2} \sum_{u \in S\cup T} \deg_{S\cup T}(u) \leq \big((1-\gamma_{\mathcal{G}})\sigma\tau + \gamma_{\mathcal{G}}\sqrt{\sigma\tau}\big)\, \Delta n \ .$$

Combining the last two equations yields

$$\max\{\delta_A\sigma, \delta_B\tau\} \leq (1-\gamma_{\mathcal{G}})\sigma\tau + \gamma_{\mathcal{G}}\sqrt{\sigma\tau} \ . \tag{2.7}$$

We now distinguish between two cases.

Case 1: $\sigma/\tau \leq \delta_B/\delta_A$. Here (2.7) becomes

$$\delta_B\tau \leq (1-\gamma_{\mathcal{G}})\sigma\tau + \gamma_{\mathcal{G}}\sqrt{\sigma\tau}$$

and, so,

$$\sigma \geq \frac{\delta_B - \gamma_{\mathcal{G}}\sqrt{\sigma/\tau}}{1-\gamma_{\mathcal{G}}} \geq \frac{\delta_B - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}}{1-\gamma_{\mathcal{G}}} \ . \tag{2.8}$$

Case 2: $\sigma/\tau > \delta_B/\delta_A$. By exchanging between σ and τ and between δ_A and δ_B in (2.8), we get

$$\tau \geq \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_B}}{1-\gamma_{\mathcal{G}}} \ .$$

Therefore,

$$\sigma > \frac{\delta_B}{\delta_A} \cdot \tau \geq \frac{\delta_B}{\delta_A} \cdot \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_B}}{1-\gamma_{\mathcal{G}}} = \frac{\delta_B - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}}{1-\gamma_{\mathcal{G}}} \ .$$

Either case yields the desired lower bound on the size, σn, of the support S of $\boldsymbol{x}$. □

The next example demonstrates how the parameters of $\mathbb{C}_\Phi$ can be tuned so that the improvement (2.1) of property (P1) holds.

Example 2.2.1 Fix $\delta_A = \epsilon$ for some small $\epsilon \in (0,1]$ (in which case $r_A > 1-\epsilon$), and then select q and Δ so that $q > \Delta \geq 4/\epsilon^3$. For such parameters, we can take $\mathcal{C}_A$ and $\mathcal{C}_B$ to be $[\Delta, r_A\Delta, \delta_A\Delta]$ and $[\Delta, r_B\Delta, \delta_B\Delta]$ GRS codes over $\mathbb{F}$, respectively (with $r_A+\delta_A = r_B+\delta_B = 1$). We also assume that $\mathcal{G}$ is a Ramanujan bipartite graph, in which case

$$\gamma_{\mathcal{G}} \leq \frac{2\sqrt{\Delta-1}}{\Delta} < \epsilon^{3/2} \ .$$

By (2.6), the rate of $\mathbb{C}_\Phi$ is bounded from below by

$$1 - \frac{1}{1-\epsilon} + \frac{r_B}{1-\epsilon} > r_B - \epsilon \; ,$$

and by Theorem 2.2.1, the relative minimum distance is at least

$$\begin{aligned} \frac{\delta_B - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}}{1-\gamma_{\mathcal{G}}} &\geq \delta_B - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A} > \delta_B - \epsilon^{3/2} \cdot \frac{1}{\sqrt{\epsilon}} \\ &= \delta_B - \epsilon > 1{-}r_B{-}\epsilon \; . \end{aligned}$$

Thus, the code $\mathbb{C}_\Phi$ approaches the Singleton bound when $\epsilon \to 0$. In addition, if q and Δ are selected to be (no larger than) $O(1/\epsilon^3)$, then the alphabet Φ has size

$$|\Phi| = q^{r_A \Delta} = 2^{O\left((\log(1/\epsilon))/\epsilon^3\right)} \; .$$

□

From Example 2.2.1 we can state the following corollary.

Corollary 2.2.4 *For any design rate $\mathcal{R} \in (0,1]$ and sufficiently small $\epsilon > 0$ there is an infinite family of codes $\mathbb{C}_\Phi$ of rate at least $\mathcal{R}$ and relative minimum distance greater than $1 - \mathcal{R} - \epsilon$, over an alphabet of size as in (2.1).*

2.3 Decoding algorithm

Let $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ be defined over $\mathbb{F} = \mathrm{GF}(q)$ as in Section 2.1. Figure 2.1 presents an adaptation of the iterative decoder of Sipser and Spielman [79] and Zémor [84] to the code $\mathbb{C}_\Phi$, with the additional feature of handling erasures (as well as errors over Φ): as we show in Theorem 2.3.1 below, the algorithm corrects any pattern of ϑ errors and ρ erasures, provided that $\vartheta + (\rho/2) < \beta n$, where

$$\beta = \frac{(\delta_B/2) - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}}{1-\gamma_{\mathcal{G}}} \; .$$

Note that β equals approximately half the lower bound in Theorem 2.2.1. The value of ν in the algorithm, which is specified in Theorem 2.3.1 below, grows logarithmically with n.

We use the notation "?" to stand for an erasure. The algorithm in Figure 2.1 makes use of a word $\boldsymbol{z} = (z_e)_{e \in E}$ over $\mathbb{F} \cup \{?\}$ that is initialized according to the contents of the received word $\boldsymbol{y}$ as follows. Each sub-block $(\boldsymbol{z})_{E(u)}$ that corresponds to a non-erased entry $\boldsymbol{y}_u$ of $\boldsymbol{y}$ is initialized to the codeword $\mathcal{E}(\boldsymbol{y}_u)$ of $\mathcal{C}_A$. The remaining sub-blocks $(\boldsymbol{z})_{E(u)}$ are initialized as erased words of length Δ. Iterations $i = 3, 5, 7, \ldots$ use an error-correcting

```
Input:      Received word $\boldsymbol{y} = (\boldsymbol{y}_u)_{u \in A}$ in $(\Phi \cup \{?\})^n$.
Initialize: For $u \in A$ do:   $(\boldsymbol{z})_{E(u)} \leftarrow \begin{cases} \mathcal{E}(\boldsymbol{y}_u) & \text{if } \boldsymbol{y}_u \in \Phi \\ ??\ldots? & \text{if } \boldsymbol{y}_u = \,? \end{cases}$ .
Iterate:    For $i = 2, 3, \ldots, \nu$ do:
              (a) If $i$ is odd then $U \equiv A$ and $\mathcal{D} \equiv \mathcal{D}_A$, else $U \equiv B$ and $\mathcal{D} \equiv \mathcal{D}_B$.
              (b) For every $u \in U$ do: $(\boldsymbol{z})_{E(u)} \leftarrow \mathcal{D}\left((\boldsymbol{z})_{E(u)}\right)$.
Output:     $\psi_{\mathcal{E}}(\boldsymbol{z})$ if $\boldsymbol{z} \in \mathbb{C}$ (and declare 'error' otherwise).
```

Figure 2.1: Decoder for the nearly-MDS code $\mathbb{C}_\Phi$.

decoder $\mathcal{D}_A : \mathbb{F}^\Delta \rightarrow \mathcal{C}_A$ that recovers correctly any pattern of less than $\delta_A \Delta/2$ errors (over $\mathbb{F}$), and iterations $i = 2, 4, 6, \ldots$ use a combined error-erasure decoder $\mathcal{D}_B : (\mathbb{F} \cup \{?\})^\Delta \rightarrow \mathcal{C}_B$ that recovers correctly any pattern of a errors and b erasures, provided that $2a + b < \delta_B \Delta$ (b will be positive only when $i = 2$).

Theorem 2.3.1 *Suppose that*

$$\sqrt{\delta_A \delta_B} > 2\gamma_{\mathcal{G}} > 0 \ , \tag{2.9}$$

and fix σ to be a positive real number such that

$$\sigma < \beta = \frac{(\delta_B/2) - \gamma_{\mathcal{G}} \sqrt{\delta_B/\delta_A}}{1 - \gamma_{\mathcal{G}}} \ . \tag{2.10}$$

If

$$\nu = 2 \left\lfloor \log \left(\frac{\beta\sqrt{\sigma n} - \sigma}{\beta - \sigma} \right) \right\rfloor + 3$$

then the decoder in Figure 2.1 recovers correctly any pattern of ϑ errors (over Φ) and ρ erasures, provided that

$$\vartheta + \frac{\rho}{2} \leq \sigma n \ . \tag{2.11}$$

The proof of the theorem makes use of the following lemma.

Lemma 2.3.2 *Let χ, σ, and τ be as in Lemma 2.2.2, and suppose that the restriction of χ to B is not identically zero and that $\gamma_{\mathcal{G}} > 0$. Let δ_B be a real number for which the following condition is satisfied for every $v \in B$:*

$$\chi(v) > 0 \implies \sum_{u \in \mathcal{N}(v)} \chi(u) \geq \frac{\delta_B \Delta}{2} .$$

Then

$$\sqrt{\frac{\sigma}{\tau}} \geq \frac{(\delta_B/2) - (1-\gamma_{\mathcal{G}})\sigma}{\gamma_{\mathcal{G}}} .$$

The proof of Lemma 2.3.2 can be found in Appendix A. This lemma implies an upper bound on τ, in terms of σ; it can be verified that this bound is always at least as tight as Lemma 5 in [84].

Proof of Theorem 2.3.1. For $i \geq 2$, let U_i be the value of the set U at the end of iteration i in Figure 2.1, and let S_i be the set of all vertices $u \in U_i$ such that $(\boldsymbol{z})_{E(u)}$ is in error at the end of that iteration. Let $\chi_1 : (A \cup B) \to \{0, \frac{1}{2}, 1\}$ be the function

$$\chi_1(u) = \begin{cases} 1 & \text{if } u \in A \text{ and } \boldsymbol{y}_u \text{ is in error} \\ \frac{1}{2} & \text{if } u \in A \text{ and } \boldsymbol{y}_u \text{ is an erasure} \\ 0 & \text{otherwise} \end{cases} ,$$

and, for $i \geq 2$ define the function $\chi_i : (A \cup B) \to \{0, \frac{1}{2}, 1\}$ recursively by

$$\chi_i(u) = \begin{cases} 1 & \text{if } u \in S_i \\ 0 & \text{if } u \in U_i \setminus S_i \\ \chi_{i-1}(u) & \text{if } u \in U_{i-1} \end{cases} ,$$

where $U_1 = A$.

Denote

$$\sigma_i = \frac{1}{n} \sum_{u \in U_i} \chi_i(u) .$$

Obviously, $\sigma_1 n = \vartheta + (\rho/2)$ and, so, by (2.11) we have $\sigma_1 \leq \sigma$.

Let ℓ be the smallest positive integer (possibly ∞) such that $\sigma_\ell = 0$. Since both $\mathcal{D}_A$ and $\mathcal{D}_B$ are bounded-distance decoders, a vertex $v \in U_i$ can belong to S_i for even $i \geq 2$, only if the sum $\sum_{u \in \mathcal{N}(v)} \chi_i(u)$ (which equals the sum $\sum_{u \in \mathcal{N}(v)} \chi_{i-1}(u)$) is at least $\delta_B \Delta/2$. Similarly, a vertex $v \in U_i$ belongs to S_i for odd $i > 1$, only if $\sum_{u \in \mathcal{N}(v)} \chi_i(u) \geq \delta_A \Delta/2$. It follows that the function χ_i satisfies the conditions of Lemma 2.3.2 (with δ_A taken instead of δ_B for odd i) and, so,

$$\sqrt{\frac{\sigma_{i-1}}{\sigma_i}} \geq \begin{cases} \dfrac{\delta_B}{2\gamma_{\mathcal{G}}} - \dfrac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}} \sigma_{i-1} & \text{for even } 0 < i < \ell \\ \dfrac{\delta_A}{2\gamma_{\mathcal{G}}} - \dfrac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}} \sigma_{i-1} & \text{for odd } 1 < i < \ell \end{cases} . \qquad (2.12)$$

Using the condition $\sigma_1 \le \sigma < \beta$, it can be verified by induction on $i \ge 2$ that

$$\frac{\sigma_{i-1}}{\sigma_i} \ge \begin{cases} \delta_B/\delta_A & \text{for even } 0 < i < \ell \\ \delta_A/\delta_B & \text{for odd } 1 < i < \ell \end{cases} . \tag{2.13}$$

Hence, for every $i > 2$,

$$\frac{\sigma_{i-2}}{\sigma_i} = \frac{\sigma_{i-2}}{\sigma_{i-1}} \cdot \frac{\sigma_{i-1}}{\sigma_i} \ge \frac{\delta_B}{\delta_A} \cdot \frac{\delta_A}{\delta_B} = 1 ;$$

in particular, $\sigma_i \le \sigma$ for odd i and $\sigma_i \le \sigma_2$ for even i. Incorporating these inequalities into (2.12) yields

$$\frac{1}{\sqrt{\sigma_i}} \ge \frac{\delta_B}{2\gamma_{\mathcal{G}}\sqrt{\sigma_{i-1}}} - \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\sqrt{\sigma} \qquad \text{for even } 0 < i < \ell \tag{2.14}$$

and

$$\frac{1}{\sqrt{\sigma_i}} \ge \frac{\delta_A}{2\gamma_{\mathcal{G}}\sqrt{\sigma_{i-1}}} - \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\sqrt{\sigma_2} \qquad \text{for odd } 1 < i < \ell\,. \tag{2.15}$$

By combining (2.14) and (2.15) we get that for even $i > 0$,

$$\begin{aligned} \frac{2\gamma_{\mathcal{G}}}{\delta_A\sqrt{\sigma_{i+1}}} + \frac{2(1-\gamma_{\mathcal{G}})}{\delta_A}\sqrt{\sigma_2} \;&\ge\; \frac{1}{\sqrt{\sigma_i}} \\ &\ge\; \frac{\delta_B}{2\gamma_{\mathcal{G}}\sqrt{\sigma_{i-1}}} - \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\sqrt{\sigma}\,, \end{aligned}$$

or

$$\begin{aligned} \frac{1}{\sqrt{\sigma_{i+1}}} &\ge \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2\sqrt{\sigma_{i-1}}} - \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\left(\frac{\delta_A\sqrt{\sigma}}{2\gamma_{\mathcal{G}}} + \sqrt{\sigma_2}\right) \\ &\ge \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2\sqrt{\sigma_{i-1}}} - \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\left(\frac{\delta_A}{2\gamma_{\mathcal{G}}} + \sqrt{\frac{\delta_A}{\delta_B}}\right)\sqrt{\sigma} \\ &= \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\left(\frac{1}{\sqrt{\sigma_{i-1}}} - \frac{\sqrt{\sigma}}{\beta}\right) + \frac{\sqrt{\sigma}}{\beta}\,, \end{aligned} \tag{2.16}$$

where the second inequality follows from $\sigma_2 \le \sigma \cdot \delta_A/\delta_B$ (see (2.13)), and the (last) equality follows from the next chain of equalities:

$$\begin{aligned} \frac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\left(\frac{\delta_A}{2\gamma_{\mathcal{G}}} + \sqrt{\frac{\delta_A}{\delta_B}}\right)\sqrt{\sigma} &= \frac{1-\gamma_{\mathcal{G}}}{2\gamma_{\mathcal{G}}^2}\left(2\gamma_{\mathcal{G}} + \sqrt{\delta_A\delta_B}\right)\sqrt{\frac{\sigma\delta_A}{\delta_B}} \\ &= -\frac{1-\gamma_{\mathcal{G}}}{2\gamma_{\mathcal{G}}^2} \cdot \frac{4\gamma_{\mathcal{G}}^2 - \delta_A\delta_B}{\sqrt{\delta_A\delta} - 2\gamma_{\mathcal{G}}}\sqrt{\frac{\sigma\delta_A}{\delta_B}} \\ &= -\left(1 - \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)\frac{(1-\gamma_{\mathcal{G}})\sqrt{\sigma}}{(\delta_B/2) - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}} \\ &= -\left(1 - \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)\frac{\sqrt{\sigma}}{\beta}\,. \end{aligned}$$

Consider the following first-order linear recurring sequence $(\Lambda_j)_{j\geq 0}$ that satisfies

$$\Lambda_{j+1} \quad = \quad \frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\left(\Lambda_j - \frac{\sqrt{\sigma}}{\beta}\right) + \frac{\sqrt{\sigma}}{\beta} \, , \quad j \geq 0 \, ,$$

where $\Lambda_0 = 1/\sqrt{\sigma}$. From (2.16) we have $1/\sqrt{\sigma_{i+1}} \geq \Lambda_{i/2}$ for even $i \geq 0$. By solving the recurrence for (Λ_j), we obtain

$$\frac{1}{\sqrt{\sigma_{i+1}}} \; \geq \; \Lambda_{i/2} = \left(\left(\frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)^{i/2}\left(1 - \frac{\sigma}{\beta}\right) + \frac{\sigma}{\beta}\right)\frac{1}{\sqrt{\sigma}} \, . \tag{2.17}$$

From the condition (2.9) we thus get that σ_{i+1} decreases exponentially with (even) i. A sufficient condition for ending the decoding correctly after ν iterations is having $\sigma_\nu < 1/n$, or

$$\frac{1}{\sqrt{\sigma_\nu}} > \sqrt{n} \, .$$

We require therefore that ν be such that

$$\frac{1}{\sqrt{\sigma_\nu}} \geq \left(\left(\frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)^{(\nu-1)/2}\left(1 - \frac{\sigma}{\beta}\right) + \frac{\sigma}{\beta}\right)\frac{1}{\sqrt{\sigma}} > \sqrt{n} \, .$$

The latter inequality can be rewritten as

$$\left(\frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)^{(\nu-1)/2} > \frac{\sqrt{n\sigma} - (\sigma/\beta)}{1 - (\sigma/\beta)} = \frac{\beta\sqrt{n\sigma} - \sigma}{\beta - \sigma} \, ,$$

thus yielding

$$\nu > 2\log\left(\frac{\beta\sqrt{n\sigma} - \sigma}{\beta - \sigma}\right) + 1 \, ,$$

where the base of the logarithm equals $(\delta_A\delta_B)/(4\gamma_{\mathcal{G}}^2)$. In summary, the decoding will end with the correct codeword after

$$\nu = 2\left\lfloor \log\left(\frac{\beta\sqrt{n\sigma} - \sigma}{\beta - \sigma}\right)\right\rfloor + 3 \, ,$$

iterations (where the base of the logarithm again equals $(\delta_A\delta_B)/(4\gamma_{\mathcal{G}}^2)$.) □

In Lemma B.1, which appears in Appendix B, it is shown that the number of actual applications of the decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ in the algorithm in Figure 2.1 can be bounded from above by $\omega \cdot n$, where

$$\omega = 2 \cdot \left\lceil \frac{\log\left(\dfrac{\Delta\beta\sqrt{\sigma}}{\beta - \sigma}\right)}{\log\left(\dfrac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)} \right\rceil + \frac{1 + \dfrac{\delta_A}{\delta_B}}{1 - \left(\dfrac{4\gamma_{\mathcal{G}}^2}{\delta_A\delta_B}\right)^2} \, .$$

Thus, if δ_A and δ_B are fixed and the ratio σ/β is bounded away from 1 and $\mathcal{G}$ is a Ramanujan graph, then the value of ω is bounded from above by an absolute constant (independent of Δ).

The algorithm in Figure 2.1 allows us to use GMD decoding in cases where $\mathbb{C}_\Phi$ is used as an outer code in a concatenated code. In such a concatenated code, the size of the inner code is $|\Phi|$ and, thus, it does not grow with the length n of $\mathbb{C}_\Phi$. A GMD decoder will apply the algorithm in Figure 2.1 a number of times that is proportional to the minimum distance of the inner code. Thus, if the inner code has rate that is bounded away from zero, then the GMD decoder will have time complexity that grows linearly with the overall code length. Furthermore, if $\mathcal{C}_A$, $\mathcal{C}_B$, and the inner code are codes that have a polynomial-time bounded-distance decoder—e.g., if they are GRS codes—then the multiplying constant in the linear expression of the time complexity (when measured in operations in $\mathbb{F}$) is $\textsc{Poly}(\Delta)$. For the choice of parameters in Example 2.2.1, this constant is $\textsc{Poly}(1/\epsilon)$ and, since $\mathbb{F}$ is chosen in that example to have size $O(1/\epsilon^3)$, each operation in $\mathbb{F}$ can in turn be implemented by $\textsc{Poly}(\log(1/\epsilon))$ bit operations. (We remark that in all our complexity estimates, we assume that the graph $\mathcal{G}$ is "hard-wired" so that we can ignore the complexity of figuring out the set of incident edges of a given vertex in $\mathcal{G}$. Along these lines, we assume that each access to an entry takes constant time, even though the length of the index of that entry may grow logarithmically with the code length. See the discussion in [79, Section II].)

When the inner code is taken as $\mathcal{C}_A$, the concatenation results in the code $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ (of length Δn) over $\mathbb{F}$, and the (linear-time) correctable fraction of errors is then the product $\delta_A \cdot \sigma$, for any positive real σ that satisfies (2.10). A special case of this result, for $\mathbb{F} = \mathrm{GF}(2)$ and $\mathcal{C}_A = \mathcal{C}_B$, was presented in our earlier work [80], yet the analysis therein was different. A linear-time decoder for $\mathbb{C}$ was also presented by Barg and Zémor in [8], except that their decoder requires finding a codeword that minimizes some weighted distance function, and we are unaware of a method that performs this task in time complexity that is $\textsc{Poly}(\Delta)$—even when $\mathcal{C}_A$ and $\mathcal{C}_B$ have a polynomial-time bounded-distance decoder.

2.4 Construction which is also linear-time encodable

In this section, we use the construction $\mathbb{C}_\Phi$ of Section 2.1 as a building block in obtaining a second construction, which satisfies all properties (P1)–(P3) over an alphabet whose size is given by (2.1).

2.4.1 Outline of the construction

Let $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ be defined over $\mathbb{F} = \mathrm{GF}(q)$ as in Section 2.1. The first simple observation that provides the intuition behind the upcoming construction is that the encoding of $\mathbb{C}$, and hence of $\mathbb{C}_\Phi$, can be easily implemented in linear time if the code $\mathcal{C}_A$ has rate $r_A = 1$, in which case $\Phi = \mathbb{F}^{\Delta}$. The definition of $\mathbb{C}$ then reduces to

$$\mathbb{C} = \left\{ \boldsymbol{c} \in \mathbb{F}^{|E|} \,:\, (\boldsymbol{c})_{E(v)} \in \mathcal{C}_B \text{ for every } v \in B \right\} \,.$$

We can implement an encoder of $\mathbb{C}$ as follows. Let $\mathcal{E}_B : \mathbb{F}^{r_B\Delta} \to \mathcal{C}_B$ be some one-to-one encoding mapping of $\mathcal{C}_B$. Given an information word $\boldsymbol{\eta}$ in $\mathbb{F}^{r_B\Delta n}$, it is first recast into a word of length n over $\mathbb{F}^{r_B\Delta}$ by sub-dividing it into sub-blocks $\boldsymbol{\eta}_v \in \mathbb{F}^{r_B\Delta}$ that are indexed by $v \in B$; then a codeword $\boldsymbol{c} \in \mathbb{C}$ is computed by

$$(\boldsymbol{c})_{E(v)} = \mathcal{E}_B(\boldsymbol{\eta}_v) \,, \quad v \in B \,.$$

By selecting $\mathcal{E}$ in (2.3) as the identity mapping, we get that the respective codeword $\boldsymbol{x} = (\boldsymbol{x}_u)_{u\in A} = \psi_{\mathcal{E}}(\boldsymbol{c})$ in $\mathbb{C}_\Phi$ is

$$\boldsymbol{x}_u = (\boldsymbol{c})_{E(u)} \,, \quad u \in A \,.$$

Thus, each of the Δ entries (over $\mathbb{F}$) of the sub-block $\boldsymbol{x}_u$ can be associated with a vertex $v \in \mathcal{N}(u)$, and the value assigned to that entry is equal to one of the entries in $\mathcal{E}_B(\boldsymbol{\eta}_v)$.

While having $\mathcal{C}_A = \Phi$ $(= \mathbb{F}^{\Delta})$ allows easy encoding, the minimum distance of the resulting code $\mathbb{C}_\Phi$ is obviously poor. To resolve this problem, we insert into the construction another linear $[\Delta, r_0\Delta, \delta_0\Delta]$ code $\mathcal{C}_0$ over $\mathbb{F}$. Let H_0 be some $((1-r_0)\Delta) \times \Delta$ parity-check matrix of $\mathcal{C}_0$ and for a vector $\boldsymbol{h} \in \mathbb{F}^{(1-r_0)\Delta}$, denote by $\mathcal{C}_0(\boldsymbol{h})$ the following coset of $\mathcal{C}_0$ within Φ:

$$\mathcal{C}_0(\boldsymbol{h}) = \{\boldsymbol{v} \in \Phi \,:\, H_0\boldsymbol{v} = \boldsymbol{h}\} \,.$$

Fix now a list of vectors $\boldsymbol{s} = (\boldsymbol{h}_u)_{u\in A}$ where $\boldsymbol{h}_u \in \mathbb{F}^{(1-r_0)\Delta}$, and define the subset $\mathbb{C}(\boldsymbol{s})$ of $\mathbb{C}$ by

$$\mathbb{C}(\boldsymbol{s}) = \{\boldsymbol{c} \in \mathbb{C} \,:\, (\boldsymbol{c})_{E(u)} \in \mathcal{C}_0(\boldsymbol{h}_u) \text{ for every } u \in A\} \,;$$

accordingly, define the subset $(\mathbb{C}(\boldsymbol{s}))_\Phi$ of $\mathbb{C}_\Phi$ by

$$(\mathbb{C}(\boldsymbol{s}))_\Phi = \left\{ \psi_{\mathcal{E}}(\boldsymbol{c}) = \left((\boldsymbol{c})_{E(u)}\right)_{u\in A} \,:\, \boldsymbol{c} \in \mathbb{C}(\boldsymbol{s}) \right\} \,.$$

Now, if $\boldsymbol{s}$ is all-zero, then $\mathbb{C}(\boldsymbol{s})$ coincides with the code $\mathbb{C}(\boldsymbol{0}) = (\mathcal{G}, \mathcal{C}_0 : \mathcal{C}_B)$; otherwise, $\mathbb{C}(\boldsymbol{s})$ is either empty or is a coset of $\mathbb{C}(\boldsymbol{0})$, where $\mathbb{C}(\boldsymbol{0})$ is regarded as a linear subspace of $\mathbb{C}$ over $\mathbb{F}$. From this observation we conclude that the lower bound in Theorem 2.2.1 applies to any nonempty subset $(\mathbb{C}(\boldsymbol{s}))_\Phi$, except that we need to replace δ_A by δ_0.

In addition, a simple modification in the algorithm in Figure 2.1 adapts it to decode $(\mathbb{C}(\boldsymbol{s}))_\Phi$ so that Theorem 2.3.1 holds (again under the change $\delta_A \leftrightarrow \delta_0$): during odd iterations i, we apply to each sub-block $(\boldsymbol{z})_{E(u)}$ a bounded-distance decoder of $\mathcal{C}_0(\boldsymbol{h}_u)$, instead of the decoder $\mathcal{D}_A$.

Therefore, our strategy in designing the linear-time encodable codes will be as follows. The raw data will first be encoded into a codeword $\boldsymbol{c}$ of $\mathbb{C}$ (where $\mathcal{C}_A = \Phi$). Then we compute the n vectors

$$\boldsymbol{h}_u = H_0 \cdot (\boldsymbol{c})_{E(u)} \,, \quad u \in A \,,$$

and produce the list $\boldsymbol{s} = (\boldsymbol{h}_u)_{u \in A}$; clearly, $\boldsymbol{c}$ belongs to $\mathbb{C}(\boldsymbol{s})$. The list $\boldsymbol{s}$ will then undergo additional encoding stages, and the result will be merged with $\psi_{\mathcal{E}}(\boldsymbol{c})$ to produce the final codeword. The parameters of $\mathcal{C}_0$, which determine the size of $\boldsymbol{s}$, will be chosen so that the overhead due to $\boldsymbol{s}$ will be negligible.

During decoding, $\boldsymbol{s}$ will be recovered first, and then we will apply the aforementioned adaptation to $(\mathbb{C}(\boldsymbol{s}))_{\Phi}$ of the decoder in Figure 2.1, to reconstruct the information word $\boldsymbol{\eta}$.

2.4.2 Details of the construction

We now describe the construction in more detail. We let $\mathbb{F}$ be the field $\mathrm{GF}(q)$ and Δ_1 and Δ_2 be positive integers. The construction makes use of two bipartite regular graphs,

$$\mathcal{G}_1 = (A : B, E_1) \qquad \text{and} \qquad \mathcal{G}_2 = (A : B, E_2) \ ,$$

of degrees Δ_1 and Δ_2, respectively. Both graphs have the same number of vertices; in fact, we are making a stronger assumption whereby both graphs are defined over the same set of vertices. We denote by n the size of A (or B) and by Φ_1 and Φ_2 the alphabets $\mathbb{F}^{\Delta_1}$ and $\mathbb{F}^{\Delta_2}$, respectively. The notations $E_1(u)$ and $E_2(u)$ will stand for the sets of edges that are incident with a vertex u in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively.

We also assume that we have at our disposal the following four codes:

- a linear $[\Delta_1, r_0\Delta_1, \delta_0\Delta_1]$ code $\mathcal{C}_0$ over $\mathbb{F}$;
- a linear $[\Delta_1, r_1\Delta_1, \delta_1\Delta_1]$ code $\mathcal{C}_1$ over $\mathbb{F}$;
- a linear $[\Delta_2, r_2\Delta_2, \delta_2\Delta_2]$ code $\mathcal{C}_2$ over $\mathbb{F}$;
- a code $\mathcal{C}_{\mathrm{m}}$ of length n and rate r_{m} over the alphabet $\Phi_{\mathrm{m}} = \mathbb{F}^{r_2\Delta_2}$.

The rates of these codes need to satisfy the relation

$$(1{-}r_0)\Delta_1 = r_{\mathrm{m}} r_2 \Delta_2 \ ,$$

and the code $\mathcal{C}_{\mathrm{m}}$ is assumed to have the following properties:

1. Its rate is bounded away from zero: there is a universal positive constant κ such that $r_{\mathrm{m}} \geq \kappa$.
2. $\mathcal{C}_{\mathrm{m}}$ is linear-time encodable, and the encoding time per symbol is $\textsc{Poly}(\log |\Phi_{\mathrm{m}}|)$.
3. $\mathcal{C}_{\mathrm{m}}$ has a decoder that recovers in linear-time any pattern of up to μn errors (over the alphabet Φ_{m}), where μ is a universal positive constant. The time complexity per symbol of the decoder is $\textsc{Poly}(\log |\Phi_{\mathrm{m}}|)$.

(By a universal constant we mean a value that does not depend on any other parameter, not even on the size of Φ_{m}.) For example, we can select as $\mathcal{C}_{\mathrm{m}}$ the code of Spielman in [81], in which case κ can be taken as $1/4$.

Based on these ingredients, we introduce the codes

$$\mathbb{C}_1 = (\mathcal{G}_1, \Phi_1 : \mathcal{C}_1) \qquad \text{and} \qquad \mathbb{C}_2 = (\mathcal{G}_2, \Phi_2 : \mathcal{C}_2)$$

over $\mathbb{F}$. The code $\mathbb{C}_1$ will play the role of the code $\mathbb{C}$ as outlined in Section 2.4.1, whereas the codes $\mathcal{C}_{\mathrm{m}}$ and $\mathbb{C}_2$ will be utilized for the encoding of the list $\boldsymbol{s}$ that was described there.

The overall construction, which we denote by $\mathbb{C}$, is now defined as the set of all words of length n over the alphabet

$$\Phi = \Phi_1 \times \Phi_2$$

that are obtained by applying the encoding algorithm in Figure 2.2 to information words $\boldsymbol{\eta}$ of length n over $\mathbb{F}^{r_1 \Delta_1}$. A schematic diagram of the algorithm is shown in Figure 2.3. (In this algorithm, we use a notational convention whereby entries of information words $\boldsymbol{\eta}$ are indexed by B, and so are codewords of $\mathcal{C}_{\mathrm{m}}$.)

From the discussion in Section 2.4.1 and from the assumption on the code $\mathcal{C}_{\mathrm{m}}$ it readily follows that the encoder in Figure 2.2 can be implemented in linear time, where the encoding complexity per symbol (when measured in operations in $\mathbb{F}$) is $\textsc{Poly}(\Delta_1, \Delta_2)$. The rate of $\mathbb{C}$ is also easy to compute: the encoder in Figure 2.2 maps, in a one-to-one manner, an information word of length n over an alphabet of size $q^{r_1 \Delta_1}$, into a codeword of length n over an alphabet Φ of size $q^{\Delta_1 + \Delta_2}$. Thus, the rate of $\mathbb{C}$ is

$$\frac{r_1 \Delta_1 n}{(\Delta_1 + \Delta_2) n} = \frac{r_1}{1 + (\Delta_2/\Delta_1)} \; . \tag{2.18}$$

In the next section, we show how the parameters of $\mathbb{C}$ can be selected so that it becomes nearly-MDS and also linear-time decodable.

2.4.3 Design, decoding, and analysis

We will select the parameters of $\mathbb{C}$ quite similarly to Example 2.2.1. We assume that the rates r_1 and r_2 of $\mathcal{C}_1$ and $\mathcal{C}_2$ are the same and are equal to some prescribed value $\mathcal{R}$, and define

$$\alpha_{\mathcal{R}} = 8 \cdot (1{-}\mathcal{R}) \cdot \max\{\mathcal{R}/\mu, 2/\kappa\}$$

(notice that $\alpha_{\mathcal{R}}$ can be bounded from above by a universal constant that does not depend on $\mathcal{R}$, e.g., by $16/\min\{2\mu, \kappa\}$). We set $\delta_0 = \kappa \cdot \epsilon$ for some positive $\epsilon < \mathcal{R}$ (in which case $1{-}r_0 < \kappa \cdot \epsilon$), and then select q, Δ_1, and Δ_2 so that $q > \Delta_1 \ge \alpha_{\mathcal{R}}/\epsilon^3$ and

$$\Delta_2 = \frac{(1{-}r_0)\Delta_1}{r_{\mathrm{m}} \mathcal{R}} \quad (< \Delta_1) \; ; \tag{2.19}$$

Input: Information word $\boldsymbol{\eta} = (\boldsymbol{\eta}_v)_{v \in B}$ of length n over $\mathbb{F}^{r_1 \Delta_1}$.

(E1) Using an encoder $\mathcal{E}_1 : \mathbb{F}^{r_1 \Delta_1} \rightarrow \mathcal{C}_1$, map $\boldsymbol{\eta}$ into a codeword $\boldsymbol{c}$ of $\mathbb{C}_1$ by

$$(\boldsymbol{c})_{E_1(v)} \leftarrow \mathcal{E}_1(\boldsymbol{\eta}_v) \ , \quad v \in B \ .$$

(E2) Fix some $((1-r_0)\Delta_1) \times \Delta_1$ parity-check matrix H_0 of $\mathcal{C}_0$ over $\mathbb{F}$, and compute the n vectors

$$\boldsymbol{h}_u \leftarrow H_0 \cdot (\boldsymbol{c})_{E_1(u)} \ , \quad u \in A \ ,$$

to produce the list $\boldsymbol{s} = (\boldsymbol{h}_u)_{u \in A}$.

(E3) Regard $\boldsymbol{s}$ as a word of length $(1-r_0)\Delta_1 n$ $(= r_{\mathrm{m}} r_2 \Delta_2 n)$ over $\mathbb{F}$, and map it by an encoder of $\mathcal{C}_{\mathrm{m}}$ into a codeword $\boldsymbol{w} = (\boldsymbol{w}_v)_{v \in B}$ of $\mathcal{C}_{\mathrm{m}}$.

(E4) Using an encoder $\mathcal{E}_2 : \mathbb{F}^{r_2 \Delta_2} \rightarrow \mathcal{C}_2$, map $\boldsymbol{w}$ into a codeword $\boldsymbol{d}$ of $\mathbb{C}_2$ by

$$(\boldsymbol{d})_{E_2(v)} \leftarrow \mathcal{E}_2(\boldsymbol{w}_v) \ , \quad v \in B \ .$$

Output: Word $\boldsymbol{x} = (\boldsymbol{x}_u)_{u \in A}$ in $(\Phi_1 \times \Phi_2)^n$ whose components are given by the pairs

$$\boldsymbol{x}_u = \left((\boldsymbol{c})_{E_1(u)}, (\boldsymbol{d})_{E_2(u)} \right) \ , \quad u \in A \ .$$

Figure 2.2: Encoder for the code $\mathbb{C}$.

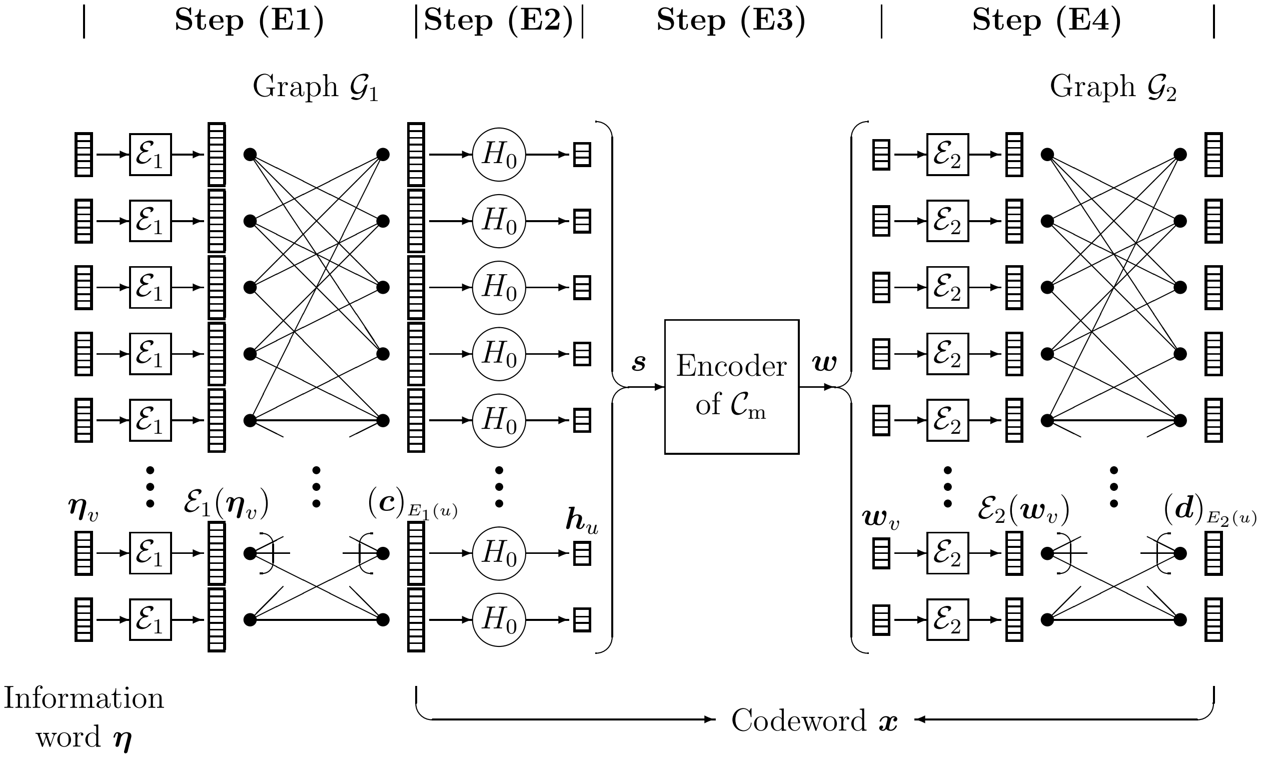
Step (E1)
Step (E2)
Step (E3)
Step (E4)
Graph $\mathcal{G}_1$
Graph $\mathcal{G}_2$
$\mathcal{E}_1$
H_0
$\boldsymbol{s}$
Encoder of $\mathcal{C}_{\mathrm{m}}$
$\boldsymbol{w}$
$\mathcal{E}_2$
$\boldsymbol{\eta}_v$
$\mathcal{E}_1(\boldsymbol{\eta}_v)$
$(\boldsymbol{c})_{E_1(u)}$
$\boldsymbol{h}_u$
$\boldsymbol{w}_v$
$\mathcal{E}_2(\boldsymbol{w}_v)$
$(\boldsymbol{d})_{E_2(u)}$
Information word $\boldsymbol{\eta}$
Codeword $\boldsymbol{x}$

Figure 2.3: Schematic diagram of the encoder for $\mathbb{C}$.

yet we also assume that q is (no larger than) $O(1/\epsilon^3)$. The graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are taken as Ramanujan graphs and $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$ are taken as GRS codes over $\mathbb{F}$. (Requiring that both Δ_1 and Δ_2 be valid degrees of Ramanujan graphs imposes some restrictions on the value $(1-r_0)/(r_{\mathrm{m}}\mathcal{R})$. These restrictions can be satisfied by tuning the precise rate of $\mathcal{C}_{\mathrm{m}}$ last.)

Given this choice of parameters, we obtain from (2.19) that $\Delta_2/\Delta_1 < \epsilon/\mathcal{R}$ and, so, the rate (2.18) of $\mathbb{C}$ is greater than

$$\frac{\mathcal{R}}{1+(\epsilon/\mathcal{R})} > \mathcal{R} - \epsilon \; . \tag{2.20}$$

The alphabet size of $\mathbb{C}$ is

$$|\Phi| = |\Phi_1| \cdot |\Phi_2| = q^{\Delta_1+\Delta_2} = 2^{O\left((\log(1/\epsilon))/\epsilon^3\right)} \; ,$$

as in (2.1), where we have absorbed into the $O(\cdot)$ term the constants κ and μ.

Our next step in the analysis of the code $\mathbb{C}$ consists of showing that there exists a linear-time decoder which recovers correctly any pattern of ϑ errors and ρ erasures, provided that

$$2\vartheta + \rho \le (1-\mathcal{R}-\epsilon)n \; . \tag{2.21}$$

This, in turn, will also imply that the relative minimum distance of $\mathbb{C}$ is greater than $1-\mathcal{R}-\epsilon$, thus establishing with (2.20) the fact that $\mathbb{C}$ is nearly-MDS.

Let $\boldsymbol{x} = (\boldsymbol{x}_u)_{u \in A}$ be the transmitted codeword of $\mathbb{C}$, where

$$\boldsymbol{x}_u = \left((\boldsymbol{c})_{E_1(u)}, (\boldsymbol{d})_{E_2(u)}\right) \; ,$$

and let $\boldsymbol{y} = (\boldsymbol{y}_u)_{u \in A}$ be the received word; each entry $\boldsymbol{y}_u$ takes the form $(\boldsymbol{y}_{u,1}, \boldsymbol{y}_{u,2})$, where $\boldsymbol{y}_{u,1} \in \Phi_1 \cup \{?\}$ and $\boldsymbol{y}_{u,2} \in \Phi_2 \cup \{?\}$. Consider the application of the algorithm in Figure 2.4 to $\boldsymbol{y}$, assuming that $\boldsymbol{y}$ contains ϑ errors and ρ erasures, where $2\vartheta + \rho \le (1-\mathcal{R}-\epsilon)n$.

Step (D1) is the counterpart of the initialization step in Figure 2.1 (the entries of $\boldsymbol{z}$ here are indexed by the edges of $\mathcal{G}_2$).

The role of Step (D2) is to compute a word $\tilde{\boldsymbol{w}} \in \Phi_{\mathrm{m}}^n$ that is close to the codeword $\boldsymbol{w}$ of $\mathcal{C}_{\mathrm{m}}$, which was generated in Step (E3) of Figure 2.2. Step (D2) uses the inverse of the encoder $\mathcal{E}_2$ (which was used in Step (E4)) and also a combined error-erasure decoder $\mathcal{D}_2 : (\mathbb{F} \cup \{?\})^{\Delta_2} \to \mathcal{C}_2$ that recovers correctly any pattern of a errors (over $\mathbb{F}$) and b erasures, provided that $2a + b < \delta_2\Delta_2$. The next lemma provides an upper bound on the Hamming distance between $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ (as words of length n over Φ_{m}).

Lemma 2.4.1 *Under the assumption (2.21), the Hamming distance between $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ (as words over Φ_{m}) is less than μn.*

Input: Received word $\boldsymbol{y} = (\boldsymbol{y}_u)_{u \in A}$ in $(\Phi \cup \{?\})^n$.

(D1) For $u \in A$ do: $(\boldsymbol{z})_{E_2(u)} \leftarrow \begin{cases} \boldsymbol{y}_{u,2} & \text{if } \boldsymbol{y}_{u,2} \in \Phi_1 \\ ??\ldots? & \text{if } \boldsymbol{y}_{u,2} = ? \end{cases}$.

(D2) For $v \in B$ do: $\tilde{\boldsymbol{w}}_v \leftarrow \mathcal{E}_2^{-1}\left(\mathcal{D}_2\left((\boldsymbol{z})_{E_2(v)}\right)\right)$.

(D3) Apply a decoder of $\mathcal{C}_{\mathrm{m}}$ to $\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{w}}_v)_{v \in B}$ to produce an information word $\hat{\boldsymbol{s}} \in \mathbb{F}^{(1-r_0)\Delta_1 n}$.

(D4) Apply a decoder for $(\mathbb{C}_1(\hat{\boldsymbol{s}}))_{\Phi_1}$ to $(\boldsymbol{y}_{u,1})_{u \in A}$, as described in Section 2.4.1, to produce an information word $\hat{\boldsymbol{\eta}} = (\hat{\boldsymbol{\eta}}_v)_{v \in B}$.

Output: Information word $\hat{\boldsymbol{\eta}} = (\hat{\boldsymbol{\eta}}_v)_{v \in B}$ of length n over $\mathbb{F}^{\mathcal{R}\Delta_1}$.

Figure 2.4: Decoder for the nearly-MDS linear-time encodable code $\mathbb{C}_\Phi$.

Proof. Define the function $\chi : (A \cup B) \to \{0, \frac{1}{2}, 1\}$ by

$$\chi(u) = \begin{cases} 1 & \text{if } u \in A \text{ and } \boldsymbol{y}_{u,2} \text{ is in error} \\ \frac{1}{2} & \text{if } u \in A \text{ and } \boldsymbol{y}_{u,2} \text{ is an erasure} \\ 1 & \text{if } u \in B \text{ and } \tilde{\boldsymbol{w}}_u \neq \boldsymbol{w}_u \\ 0 & \text{otherwise} \end{cases} .$$

Assuming that $\tilde{\boldsymbol{w}} \neq \boldsymbol{w}$, this function satisfies the conditions of Lemma 2.3.2 with respect to the graph $\mathcal{G}_2$, where σn equals $\vartheta + (\rho/2)$ and τn equals the number of vertices $v \in B$ such that $\tilde{\boldsymbol{w}}_v \neq \boldsymbol{w}_v$. By that lemma we get

$$\begin{aligned} \sqrt{\frac{\sigma}{\tau}} &\geq \frac{(\delta_2/2) - (1-\gamma_2)\sigma}{\gamma_2} \geq \frac{(\delta_2/2) - \sigma}{\gamma_2} \\ &> \frac{1-\mathcal{R} - 2\sigma}{2\gamma_2} \geq \frac{\epsilon}{2\gamma_2} , \end{aligned} \tag{2.22}$$

where γ_2 stands for $\gamma_{\mathcal{G}_2}$ and the last inequality follows from (2.21). Now, by (2.19) we have

$$\Delta_2 = \frac{(1-r_0)\Delta_1}{r_{\mathrm{m}}\mathcal{R}} > \frac{\epsilon\Delta_1}{\mathcal{R}} \geq \frac{\alpha_{\mathcal{R}}}{\mathcal{R} \cdot \epsilon^2} \geq \frac{8(1-\mathcal{R})}{\mu \cdot \epsilon^2} ,$$

from which we get the following upper bound on the square of γ_2:

$$\gamma_2^2 \leq \frac{4(\Delta_2-1)}{\Delta_2^2} < \frac{4}{\Delta_2} \leq \frac{\mu \cdot \epsilon^2}{2(1-\mathcal{R})} .$$

Combining this bound with (2.22) yields

$$\frac{\sigma}{\tau} > \frac{1-\mathcal{R}}{2\mu} \, ,$$

namely, $\tau < 2\mu\sigma/(1-\mathcal{R}) < \mu$. □

It follows from Lemma 2.4.1 that Step (D2) reduces the number of errors in $\tilde{\boldsymbol{w}}$ to the extent that allows a linear-time decoder of $\mathcal{C}_{\mathrm{m}}$ to fully recover the errors in $\tilde{\boldsymbol{w}}$ in Step (D3). Hence, the list $\hat{\boldsymbol{s}}$, which is computed in Step (D3), is identical with the list $\boldsymbol{s}$ that was originally encoded in Step (E2).

Finally, to show that Step (D4) yields complete recovery from errors, we apply Theorem 2.3.1 to the parameters of the code $(\mathcal{G}_1, \mathcal{C}_0 : \mathcal{C}_1)$. Here $\delta_0 = \kappa \cdot \epsilon$ and

$$\gamma_1 = \gamma_{\mathcal{G}_1} < \frac{2}{\sqrt{\Delta_1}} \le \frac{2\epsilon^{3/2}}{\sqrt{\alpha_{\mathcal{R}}}} \le \frac{\epsilon^{3/2}}{2\sqrt{(1-\mathcal{R})/\kappa}} \; ;$$

therefore,

$$\beta = \frac{(\delta_1/2) - \gamma_1\sqrt{\delta_1/\delta_0}}{1-\gamma_1} > \frac{1-\mathcal{R}}{2} - \gamma_1\sqrt{\frac{1-\mathcal{R}}{\delta_0}} > \frac{1-\mathcal{R}-\epsilon}{2}$$

and, so, by (2.21), the conditions of Theorem 2.3.1 hold for $\sigma = (1-\mathcal{R}-\epsilon)/2$ (note that $\beta > 0$ yields $\sqrt{\delta_0\delta_1} > 2\gamma_1$, thus (2.9) holds).

Chapter 3

Decoding Expander Codes at Rates Close to Capacity

The material in this section has been published as [6], [7].

3.1 Introduction

The speed of the decrease of the decoding error probability as a function of the code length, N, is a characteristic of capacity-approaching codes, which was widely studied for many code families. However, this probability depends also on the ratio between the channel capacity and an actual code rate. Namely, let the code rate be $\mathcal{R} = (1 - \varepsilon)\mathsf{C}$, where C is the channel capacity. It is an interesting question to ask is how the decoding error probability depends on ε.

Another characteristic of (decoding algorithms of) codes is the time complexity of decoding. As of yet, there are known families of capacity-achieving codes (over various channels) with decoding algorithm time complexity only linear in N. However, one might look on the decoding time complexity of code families in terms of ε. In the next two paragraphs we discuss these characteristics for two code families.

It is known that LDPC-type codes can attain the capacity of the binary erasure channel (BEC); the reader can refer to [54], [65], [70]. It is generally believed that LDPC-type codes can approach capacity of a variety of other communication channels. However, it is also believed that the decoding error probability decreases only polynomially with the code length. As for the decoding time complexity, it was conjectured in [45] that the per-bit complexity of message-passing decoding (e.g. [34], [69]) of LDPC or irregular repeat accumulative (IRA) codes over any 'typical' channel is $O\left(\log \frac{1}{\pi}\right) + O\left(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}\right)$, where π is the decoding error probability. Lately, for LDPC-type codes with message-passing decoding over BEC, the time complexity was shown to be linear in the code length and sub-linear in $1/\varepsilon$. More specifically, it was shown in [54] and [65] that the decoding complexity per bit for some sub-families of LDPC-type codes behaves as $O(\log(1/\varepsilon))$. Recently, in [67], IRA codes with bounded decoding complexity per bit were constructed.

In contrast, the modifications of expander codes presented in Chapter 2 and in [8], [10], [11] also attain the capacity of the memoryless q-ary symmetric channel, and the error probability decreases exponentially with the code length. Several recent works were devoted to the analysis of fraction of errors that expander codes can correct (e.g. [28], [79], [84]) and their rate-distance trade-offs (see [8], [38]). The results of this kind appear also in Chapter 2 of this thesis. While it is well known that there are decoders for expander codes having linear-time (in the code length) complexity, the dependence of this complexity on $1/\varepsilon$ was not studied. In the present chapter, we aim at studying this dependence. We investigate the time complexity of decoding algorithms of expander codes in terms of ε, in particular for the codes in [8], [10]. We show that (using known decoding algorithms) these specific codes have (per-bit) time complexity that is exponential in $1/\varepsilon^2$.

In this thesis, we study capacity-achieving codes over a binary symmetric channel (BSC). We show that if there exists a family of codes $\mathcal{C}_{in}$ of length N and rate $\mathcal{R} = (1-\varepsilon)\mathsf{C}$ (C is a BSC capacity), with the decoding probability vanishing inverse polynomially in N and ε (under conditions of our theorem), then there exists another such family of codes $\mathbb{C}_{cont}$ with the decoding error probability vanishing exponentially in N. Moreover, if the decoding time complexity of the codes $\mathcal{C}_{in}$ is polynomial in N and $1/\varepsilon$, then the decoding time complexity of the codes $\mathbb{C}_{cont}$ is linear in N and polynomial in $1/\varepsilon$.

The structure of this chapter is as follows. In Section 3.2, we describe the basic ingredients in our construction. The main result of this chapter appears in Section 3.3: we present a sufficient condition for the existence of a family of codes with the decoding error probability vanishing exponentially fast. We also analyze the decoding time complexity of the presented codes. Finally, in Sections 3.4 and 3.5, we show that the codes in [10] and [8], when decoded by known algorithms, cannot be tuned to have decoding error probability that decreases exponentially fast (in terms of N), while the respective decoding algorithms have time complexity linear in N and polynomial in $1/\varepsilon$.

3.2 Preliminaries

3.2.1 Capacity-achieving codes with fast decoding

In this section we assume the existence of some (family of) linear code $\mathcal{C}_{in}$, which achieves the capacity C of a BSC, and which has a fast decoding algorithm. We denote its rate

$\mathcal{R}_{in} = (1-\varepsilon)\mathsf{C}$, and its length n_{in} (constant for a fixed ε). Below, we discuss the parameters of this code.

Decoding complexity: we assume that the decoding complexity of $\mathcal{C}_{in}$ over the BSC is given by

$$O\left(n_{in}^{\mathsf{s}} \cdot \frac{1}{\varepsilon^{\mathsf{r}}}\right) , \tag{3.1}$$

where $\mathsf{s}, \mathsf{r} \geq 1$ are some constants. Let $\mathcal{D}_{in}$ be a decoder that has a time complexity as in (3.1).

Based on the results in [54], [65], [67], several LDPC-type code families (with respective message-passing decoding algorithms) do have such decoding complexity over the BEC (for $\mathsf{s} = 1$). There are no such results known for the BSC, although in light of the surveyed works, this assumption sounds reasonable for LDPC-type codes over the BSC.

Decoding error probability: so far, there are no satisfying results on the asymptotical behavior of the decoding error probability of LDPC-type codes over the BEC under the message-passing decoding, for rates near capacity of the BEC. The behavior of the decoding error probability of LDPC-type codes over other channels is even less investigated. In the sequel, we obtain a sufficient condition on the probability of the decoding error $\mathsf{Prob}_e(\mathcal{C}_{in})$ of the decoder $\mathcal{D}_{in}$ (for the $\mathcal{C}_{in}$) to guarantee the existence of a code with an exponentially-fast decreasing error probability.

Note: the results presented in the sequel are valid for any code $\mathcal{C}_{in}$ whose decoding time complexity and decoding error probability are as stated above. However, LDPC-type codes are very promising candidates to meet these conditions, and in fact we do not see any other candidate at the present moment. Since there is no such candidate, it makes sense to speak about LDPC-type codes in this context.

3.2.2 Nearly-MDS expander codes

In this section, we consider linear-time decodable codes of rate $1-\epsilon$ (for small $\epsilon > 0$) that can correct a fraction $\vartheta\epsilon^{\mathsf{b}}$ of errors, where $\vartheta > 0, \mathsf{b} > 0$ are constants. There are several code families known to date that can be shown to have the above property, and at the same time allow linear-time (in the code length) decoding. In this connection, the reader can refer to [8], [10], [38], [79], [84]. However, as of yet, the codes in Chapter 2 have the best relations between their rate, minimum distance and alphabet size among all known expander-based linear-time decodable codes. Moreover, unlike the codes Chapter 2, not all aforementioned codes have decoding time complexity which is polynomial in $1/\epsilon$.

In the sequel, we use the codes $\mathbb{C}_\Phi$ defined as in (2.4) with the constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$ taken as GRS codes. The parameters of the codes $\mathcal{C}_A$ and $\mathcal{C}_B$ will be defined later in this section.

Definition. An infinite sequence $\{a_i\}_{i=1}^{\infty}$, $a_i \overset{i\to\infty}{\longrightarrow} +\infty$, $a_i \in \mathbb{R}$, is called *a dense sequence of values* if $a_1 \leq 100$ and $a_{i+1} - a_i = o(a_i)$ (for $i \to \infty$). (The number 100 is a large absolute constant, the condition $a_1 \leq 100$ ensures that not all elements in the sequence are exponentially large.)

Let $\gamma_{\mathcal{G}}$ be defined as in Chapter 2. It was shown in Section 2.2 that the code $\mathbb{C}_\Phi$ has relative minimum distance as in Theorem 2.2.1, and the rate of $\mathbb{C}_\Phi$ is given in (2.6).

A linear-time decoding algorithm $\mathcal{D}_\Phi$ of $\mathbb{C}_\Phi$ was presented in Figure 2.1 in Section 2.3. It corrects any pattern of ϑ errors and ρ erasures such that $\vartheta + \frac{1}{2}\rho < \beta n$, where β is as in Theorem 2.3.1. The number of iterations m in the algorithm is such that $m = O(\log n)$.

Recall that in Figure 2.1 it is required that the decoder $\mathcal{D}_A$ is a mapping $\mathbb{F}^\Delta \to \mathcal{C}_A$ that recovers correctly any pattern of less than $\delta_A \Delta/2$ errors over $\mathbb{F}$, and the decoder $\mathcal{D}_B$ is a mapping $(\mathbb{F} \cup \{?\})^\Delta \to \mathcal{C}_B$ that recovers correctly any pattern of a errors and b erasures, provided that $2a + b < \delta_B \Delta$. The decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ are polynomial-time, for example a Berlekamp-Massey decoder can be used for both of them. It can be implemented then in time $O(\Delta^2)$.

In the next proposition, we show that the parameters of the codes $\mathbb{C}_\Phi$ of rate $1 - \epsilon$ can be tuned to correct $\vartheta'\epsilon$ errors ($\vartheta' > 0$ is a constant) for a sequence of alphabets.

Proposition 3.2.1 *For any $\epsilon \in (0, 1)$, and for a sequence of alphabets $\{\Phi_i\}_{i=1}^{\infty}$ such that the sequence $\{\log_2 |\Phi_i|\}_{i=1}^{\infty}$ is dense, the codes $\mathbb{C}_\Phi$ (as above) of rate $R_\Phi \geq 1 - \epsilon$ (with decoder $\mathcal{D}_\Phi$) can correct a fraction $\vartheta'\epsilon$ of errors, where $\vartheta' > 0$ is some constant.*

Proof. There is a dense sequence of values $\Delta \in \{\Delta_i\}_{i=1}^{\infty}$ such that there exists a family of Δ-regular bipartite Ramanujan graphs $\mathcal{G}$ (see [53], [62]). For any such value Δ, we can take both codes $\mathcal{C}_A$ and $\mathcal{C}_B$ to be GRS codes of length Δ over an alphabet of size $|\mathbb{F}| \geq \Delta$, rate $r_A = r_B = 1 - \epsilon/2$ and relative minimum distance $\delta_A = \delta_B = \epsilon/2$. Consider a code $\mathbb{C}_\Phi$ defined with respect to these $\mathcal{C}_A$ and $\mathcal{C}_B$. The rate R_Φ of $\mathbb{C}_\Phi$ satisfies $R_\Phi \geq r_A + r_B - 1 = 1 - \epsilon$. From Theorem 2.3.1, the fraction of errors that the decoder $\mathcal{D}_\Phi$ can correct is given by

$$\begin{aligned} \beta &= \frac{\delta_B/2 - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A}}{1 - \gamma_{\mathcal{G}}} \\ &\geq \epsilon/4 - \gamma_{\mathcal{G}} \\ &= \epsilon/4 - 2\sqrt{\Delta - 1}/\Delta \\ &\geq \epsilon/4 - 2/\sqrt{\Delta}\ . \end{aligned}$$

Take any Δ such that $\Delta > (16/\epsilon)^2$: for such Δ,

$$\beta > \vartheta'\epsilon\ , \text{ where } \vartheta' = 1/8\ .$$

Next, we observe that $|\Phi_i| = \Delta_i^{\Delta_i r_A}$. Based on the density of $\{\Delta_i\}_{i=1}^{\infty}$, we show that the sequence $\{\log_2 |\Phi_i|\}_{i=1}^{\infty}$ is dense as well. Indeed, for any $i \in \mathbb{N}$,

$$\begin{aligned}
&\lim_{i\to\infty} \frac{\log_2 |\Phi_{i+1}| - \log_2 |\Phi_i|}{\log_2 |\Phi_i|} \\
&= \lim_{i\to\infty} \frac{\Delta_{i+1}\log_2 \Delta_{i+1} - \Delta_i \log_2 \Delta_i}{\Delta_i \log_2 \Delta_i} \\
&= \lim_{i\to\infty} \left(\frac{\Delta_{i+1}\log_2 \Delta_{i+1}}{\Delta_i \log_2 \Delta_i} \right) - 1 \\
&= \lim_{i\to\infty} \left(\frac{\Delta_i + o(\Delta_i)}{\Delta_i} \cdot \frac{\log_2(\Delta_i + o(\Delta_i))}{\log_2 \Delta_i} \right) - 1 \\
&= 1 - 1 = 0 .
\end{aligned}$$

Finally, from [53] and [62], Δ_1 can be taken small enough, such that $\log_2 |\Phi_1| < 100$, as required. □

3.3 Main results of this chapter

3.3.1 General settings

Consider a BSC with crossover probability p, and let $\mathcal{R} = \mathsf{C}(1-\varepsilon)$ be a design rate. Take $\mathbb{F}$ to be GF(q), $q = 2^\ell$, $\ell \in \mathbb{N}$. Let $\mathcal{C}_{in}$ be a binary code of length n_{in} assumed in Section 3.2.1. It can also be seen as a linear code of length $\mathsf{n}_{in} = n_{in}/\ell$ over $(\text{GF}(2))^\ell$. Let $\mathbb{C}_\Phi$ be a linear code of length n and rate R_Φ over an alphabet $\Phi = \mathbb{F}^{\mathcal{R}_{in}\mathsf{n}_{in}}$. Pick some linear one-to-one mapping $\mathcal{E}_0 : \Phi \to \mathcal{C}_{in}$. Let $\mathbb{C}_{cont}$ be a code, corresponding to a concatenation of the code $\mathcal{C}_{in}$ (as an inner code) with the code $\mathbb{C}_\Phi$ (as an outer code), as defined in Section 1.2.7. Suppose $R_{cont} \geq \mathcal{R}$ is a rate of the (binary) code $\mathbb{C}_{cont}$ and $N_{cont} = n \cdot n_{in}$ is its length. Denote by $\mathsf{Prob}_e(\mathbb{C}_{cont})$ its error probability, under the decoding by $\mathcal{D}_{cont}$.

The following lemma is based on the result in [30, Chapter 4.2].

Lemma 3.3.1 *The error probability of the code $\mathbb{C}_{cont}$ (as defined in this section) under the decoding by $\mathcal{D}_{cont}$, when the error probability of the decoder $\mathcal{D}_{in}$ for the code $\mathcal{C}_{in}$ is $\mathsf{Prob}_e(\mathcal{C}_{in})$, and the decoder $\mathcal{D}_\Phi$ corrects any pattern of less than βn errors, is bounded by*

$$\mathsf{Prob}_e(\mathbb{C}_{cont}) \leq \exp\{-n \cdot \mathbb{E}\} = \exp\left\{ -N_{cont} \cdot \frac{\mathbb{E}}{n_{in}} \right\} ,$$

where $\mathbb{E}$ is a constant given by

$$\begin{aligned}
\mathbb{E} \;=\; & -\beta \ln\left(\mathsf{Prob}_e(\mathcal{C}_{in})\right) - (1-\beta)\ln\left(1 - \mathsf{Prob}_e(\mathcal{C}_{in})\right) \\
& + \beta \ln(\beta) + (1-\beta)\ln(1-\beta) \ .
\end{aligned} \tag{3.2}$$

If the right-hand side of (3.2) is negative, we assume that $\mathbb{E}$ is zero.

The proof of this lemma appears in Appendix A.

Remark. It is possible to improve the error exponent by a constant factor by allowing the decoder for the code $\mathcal{C}_{in}$ to output an "erasure" message in case of unreliable decoding of the code $\mathcal{C}_{in}$. See [30, Chapter 4.2] for details. We omit this analysis for the sake of simplicity.

3.3.2 Sufficient condition

In this section, we derive a sufficient condition on the probability of decoding error of the code $\mathcal{C}_{in}$ for providing a positive error exponent for the code $\mathbb{C}_{cont}$ as defined in section 3.3.1. Below, we use the notation $\mathcal{C}_{in}\left[\mathcal{R}_{in}, n_{in}\right]$ for the code $\mathcal{C}_{in}$ of rate $\mathcal{R}_{in}$ and length n_{in}.

Theorem 3.3.2 *Consider a BSC with crossover probability p and capacity $\mathsf{C} = \mathsf{C}_2(p)$. Suppose that the following two conditions hold:*

(i) There exist constants $\mathsf{b} > 0$, $\vartheta' > 0$, $\varepsilon_1 \in (0,1)$, such that for any ϵ, $0 < \epsilon < \varepsilon_1$, and for a sequence of alphabets $\{\Phi_i\}_{i=1}^{\infty}$ where the sequence $\{\log_2 |\Phi_i|\}_{i=1}^{\infty}$ is dense, there exists a family of codes $\mathbb{C}_\Phi$ of rate $1-\epsilon$ (with their respective decoders) that can correct a fraction $\vartheta' \epsilon^{\mathsf{b}}$ of errors.

(ii) There exist constants $\varepsilon_2 \in (0,1)$ and $h_0 > 0$, such that for any ϵ, $0 < \epsilon < \varepsilon_2$, the decoding error probability of a family of codes $\mathcal{C}_{in}$ satisfies

$$\mathsf{Prob}_e\left(\mathcal{C}_{in}\left[(1-\epsilon)\mathsf{C},\ \frac{1}{\epsilon^{h_0}}\right]\right) < \epsilon^{\mathsf{b}} .$$

Then, for any rate $\mathcal{R} < \mathsf{C}$, there exist a family of codes $\mathbb{C}_{cont}$ as defined in Section 3.3.1 (with their respective decoders) that has an exponentially decaying (in N_{cont}) error probability.

Proof. Let $\mathcal{R} = (1-\varepsilon)\mathsf{C}$ be a design rate of the code $\mathbb{C}_{cont}$, and $\varepsilon > 0$ be small (namely, $\varepsilon < \min\{\varepsilon_1, \varepsilon_2\}$). Let κ be a constant, $0 < \kappa < 1$, which will be defined later, and let the rate of the code $\mathcal{C}_{in}$ be $\mathcal{R}_{in} = (1-\kappa\,\varepsilon)\mathsf{C}$. We set the rate of $\mathbb{C}_\Phi$ as

$$R_\Phi = \frac{\mathcal{R}}{\mathcal{R}_{in}} = \frac{1-\varepsilon}{1-\kappa\varepsilon} = 1 - (1-\kappa)\varepsilon - \Theta(\varepsilon^2) .$$

Then, by condition (i), the fraction β of errors correctable by the code $\mathbb{C}_\Phi$ is at least $\beta \geq \vartheta'((1-\kappa)\cdot\varepsilon)^{\mathsf{b}}$.

For an alphabet Φ, the length n_{in} of the code $\mathcal{C}_{in}$ is given by

$$n_{in} = \frac{\log_2 |\Phi|}{\mathcal{R}_{in}} \ .$$

We select the smallest $\Phi \in \{\Phi_i\}_{i=1}^{\infty}$ such that

$$\log_2 |\Phi| \geq \frac{1}{(\kappa\varepsilon)^{h_0}} \ ,$$

and, so,

$$n_{in} > \frac{1}{(\kappa\varepsilon)^{h_0}} \ , \tag{3.3}$$

Next, we use Lemma 3.3.1 to evaluate the decoding error probability of the code $\mathbb{C}_{cont}$. It holds for small positive values of β that

$$(1-\beta)\ln(1-\beta) > -\beta \ ,$$

and thus, from Lemma 3.3.1 we obtain (by ignoring the positive term $-(1-\beta)\ln(1-\mathsf{Prob}_e(\mathcal{C}_{in}))$ in (3.2)),

$$\begin{aligned} \mathsf{Prob}_e\left(\mathbb{C}_{cont}\right) &< \exp\left\{-n \cdot \left(-\beta \ln\left(\mathsf{Prob}_e(\mathcal{C}_{in})\right) + \beta\ln\beta - \beta\right)\right\} \\ &= \exp\left\{-N_{cont}\frac{\beta}{n_{in}}\left(\ln\beta - \ln\left(\mathsf{Prob}_e(\mathcal{C}_{in})\right) - 1\right)\right\} \ . \end{aligned}$$

In order to have a positive error exponent, we require that

$$\ln\beta - \ln\left(\mathsf{Prob}_e(\mathcal{C}_{in})\right) - 1 > 0 \ ,$$

or, equivalently,

$$\beta > \mathsf{e} \cdot \mathsf{Prob}_e(\mathcal{C}_{in}) \ , \tag{3.4}$$

where $\mathsf{e} = 2.718\cdots$.

The decoding error probability of the selected code $\mathcal{C}_{in}$ satisfies:

$$\begin{aligned} \mathsf{Prob}_e\left(\mathcal{C}_{in}\left[(1-\kappa\varepsilon)\mathsf{C},\ n_{in}\right]\right) &< \mathsf{Prob}_e\left(\mathcal{C}_{in}\left[(1-\kappa\varepsilon)\mathsf{C},\ \frac{1}{(\kappa\varepsilon)^{h_0}}\right]\right) \\ &< (\kappa\varepsilon)^{\mathsf{b}} \ \leq \ \frac{\vartheta'((1-\kappa)\varepsilon)^{\mathsf{b}}}{\mathsf{e}} \ , \end{aligned} \tag{3.5}$$

where the first inequality is due to (3.3), the second inequality follows from condition (ii), and the third inequality can be satisfied by a selection of a small constant κ such that $\kappa^{\mathsf{b}} \leq \vartheta'(1-\kappa)^{\mathsf{b}}/\mathsf{e}$.

The inequality (3.5) implies (3.4), as required. □

Example 3.3.1 Suppose that the decoding error probability of the code $\mathcal{C}_{in}$ of rate $\mathcal{R}_{in} = (1-\varepsilon)\mathsf{C}$ and length n_{in} (for some decoder) is bounded by

$$\mathsf{Prob}_e(\mathcal{C}_{in}) < \frac{1}{n_{in}} \cdot \frac{1}{\varepsilon^4} \, .$$

We choose $h_0 = \mathsf{b}+5$ (where b is as in condition (i) of Theorem 3.3.2). There obviously exists ε_2 such that for every $0 < \epsilon < \varepsilon_2$, for a code $\mathcal{C}_{in}$ of length $n_{in} = 1/\epsilon^{h_0}$ and rate $\mathcal{R}_{in} = (1-\epsilon)\mathsf{C}$,

$$\mathsf{Prob}_e(\mathcal{C}_{in}) < \frac{1}{n_{in}} \cdot \frac{1}{\epsilon^4} = \epsilon^{h_0} \cdot \frac{1}{\epsilon^4} = \epsilon^{\mathsf{b}+1} < \epsilon^{\mathsf{b}} \, . \tag{3.6}$$

From the expression (3.6) we see that condition (ii) of Theorem 3.3.2 is satisfied. This selection yields the existence of a positive error exponent for the code $\mathbb{C}_{cont}$. □

Example 3.3.2 Suppose that the decoding error probability of the code $\mathcal{C}_{in}$ (of rate $\mathcal{R}_{in} = (1-\varepsilon)\mathsf{C}$ and length n_{in}) is bounded by

$$\mathsf{Prob}_e(\mathcal{C}_{in}) < \mathsf{e}^{-n_{in}\varepsilon^2}.$$

We choose $h_0 = 3$. There obviously exists ε_2 such that for every $0 < \epsilon < \varepsilon_2$, for the code $\mathcal{C}_{in}$ of length $n_{in} = 1/\epsilon^{h_0}$ and rate $\mathcal{R}_{in} = (1-\epsilon)\mathsf{C}$, and for every $\mathsf{b} > 0$,

$$\mathsf{Prob}_e(\mathcal{C}_{in}) < \mathsf{e}^{-n_{in}\epsilon^2} = \mathsf{e}^{-(\epsilon^2/\epsilon^3)} = \mathsf{e}^{-(1/\epsilon)} < \epsilon^{\mathsf{b}} \, ,$$

and therefore Theorem 3.3.2 yields existence of a positive error exponent for the code $\mathbb{C}_{cont}$. □

Example 3.3.3 In this example, we consider a specific case of decoding error probability for the code $\mathcal{C}_{in}$. Theorem 3.3.2 can be directly applied in this case. However, we conduct a direct minimization of the decoding error probability of the code $\mathbb{C}_{cont}$, which is obtained by concatenation of the code $\mathbb{C}_\Phi$ in Chapter 2 with the assumed code $\mathcal{C}_{in}$, and obtain an analytical expression on the error exponent. We show that the overall decoding error probability for this code $\mathbb{C}_{cont}$ has a positive error exponent.

Suppose that the decoding error probability for some inner code $\mathcal{C}_{in}$ over the BSC with crossover probability $p < \mathsf{H}_2^{-1}(1-\mathcal{R}_{in})$ and some polynomial decoder is given by:

$$\mathsf{Prob}_e(\mathcal{C}_{in}) \le \frac{1}{n_{in}^t},$$

where t is a constant, $t \ge 1$.

Below, we make a selection of parameters for the code $\mathbb{C}_{cont}$. This selection allows us to estimate the decoding error exponent as a function of ε.

Let $\mathcal{R} = (1-\varepsilon)\mathsf{C}$ be a design code rate. Pick the rate of $\mathcal{C}_{in}$ to be $\mathcal{R}_{in} = (1-\kappa\,\varepsilon)\mathsf{C}$, where $\kappa \in (0,1)$ is a constant. Then, we can write

$$\frac{\mathcal{R}}{\mathcal{R}_{in}} = \frac{\mathsf{C}(1-\varepsilon)}{\mathsf{C}(1-\kappa\,\varepsilon)} \geq 1-(1-\kappa)\varepsilon - \Theta(\varepsilon^2)\ .$$

Next, we select the parameters of the code $\mathbb{C}_\Phi$ in Chapter 2, which serves as an outer code. Take $\mathcal{C}_A$ and $\mathcal{C}_B$ as GRS codes over $\mathbb{F}$, with $|\mathbb{F}| = \Delta$. We fix $\delta_B = 1-(\mathcal{R}/\mathcal{R}_{in}) - \delta_A = \eta(1-(\mathcal{R}/\mathcal{R}_{in}))$, where $\eta \in (0,1)$ (and thus, $\delta_A = (1-\eta)(1-\mathcal{R}/\mathcal{R}_{in})$), and select the degree Δ of the graph $\mathcal{G}$ as $\Delta = \varrho/\varepsilon^2$, where ϱ is a constant such that

$$\varrho > \frac{16}{\eta(1-\eta)(1-\kappa)^2}\ .$$

We have,

$$R_\Phi \geq r_A + r_B - 1 = 1 - \delta_A - \delta_B = \mathcal{R}/\mathcal{R}_{in}\ .$$

By our selection (see (1.1)),

$$\gamma_{\mathcal{G}} \leq \frac{2}{\sqrt{\Delta}} = \frac{2\varepsilon}{\sqrt{\varrho}}\ .$$

We obtain from (2.10),

$$\beta > (\delta_B/2) - \gamma_{\mathcal{G}}\sqrt{\delta_B/\delta_A} > \vartheta'\varepsilon + o(\varepsilon)\ , \tag{3.7}$$

where

$$0 < \vartheta' = \frac{\eta(1-\kappa)}{2} - 2\sqrt{\frac{\eta}{\varrho(1-\eta)}}$$

is a constant which depends only on κ, η and ϱ.

The number of bits needed to represent each symbol of Φ is $\log_2|\Phi| = r_A\Delta \cdot \log_2|\mathbb{F}|$. Recall that $r_A = 1 - O(\varepsilon)$. Therefore, the length n_{in} of the binary code $\mathcal{C}_{in}$ is given by

$$\begin{aligned} n_{in} &= \frac{r_A\Delta}{\mathcal{R}_{in}} \cdot \log_2(\Delta) \\ &= \frac{(1-O(\varepsilon))\varrho}{\mathcal{R}_{in}\,\varepsilon^2} \cdot \log_2\left(\frac{\varrho}{\varepsilon^2}\right) \\ &= \frac{\varrho\log_2(\varrho/\varepsilon^2)}{\mathcal{R}_{in}\,\varepsilon^2} + o\left(\frac{\varrho\log_2(\varrho/\varepsilon^2)}{\mathcal{R}_{in}\,\varepsilon^2}\right)\ , \end{aligned} \tag{3.8}$$

and thus, by ignoring the small term, the decoding error probability of $\mathcal{C}_{in}$ is

$$\mathsf{Prob}_e(\mathcal{C}_{in}) \leq \left(\frac{\varepsilon^2\mathcal{R}_{in}}{\varrho\log_2(\varrho/\varepsilon^2)}\right)^t\ . \tag{3.9}$$

We substitute the expressions in (3.7) (only the main term) and (3.9) into the result of Lemma 3.3.1 to obtain

$$\begin{aligned}\mathsf{Prob}_e(\mathbb{C}_{cont}) \;&<\; \exp\Bigg\{ -n\Bigg(-\vartheta'\varepsilon\cdot t\ln\left(\frac{\varepsilon^2\mathcal{R}_{in}}{\varrho\log_2(\varrho/\varepsilon^2)}\right) \\ &\qquad - (1-\vartheta'\varepsilon)\ln\left(1-\left(\frac{\varepsilon^2\mathcal{R}_{in}}{\varrho\log_2(\varrho/\varepsilon^2)}\right)^t\right) \\ &\qquad + \vartheta'\varepsilon\ln(\vartheta'\varepsilon) + (1-\vartheta'\varepsilon)\ln(1-\vartheta'\varepsilon)\Bigg)\Bigg\}\,. \qquad (3.10)\end{aligned}$$

Note that for small $\varepsilon > 0$,

$$\ln(1-\vartheta'\varepsilon) = -\vartheta'\varepsilon + O(\varepsilon^2)\ ,$$

and

$$\ln\left(1-\left(\frac{\varepsilon^2\mathcal{R}_{in}}{\varrho\log_2(\varrho/\varepsilon^2)}\right)^t\right) = -o(\varepsilon^{2t})\ .$$

Hence Equation (3.10) (when neglecting $o(\varepsilon)$ terms) becomes

$$\begin{aligned}\mathsf{Prob}_e(\mathbb{C}_{cont}) \;&<\; \exp\left\{ -n\vartheta'\varepsilon\left(-t\ln\left(\frac{\varepsilon^2\mathcal{R}_{in}}{\varrho\log_2(\varrho/\varepsilon^2)}\right) + \ln(\vartheta'\varepsilon) - 1\right)\right\} \\ &=\; \exp\left\{-\frac{N_{cont}\vartheta'\varepsilon}{n_{in}}\cdot\ln\left(\frac{\vartheta'\varepsilon\cdot\varrho^t(\log_2(\varrho/\varepsilon^2))^t}{\mathsf{e}\cdot\varepsilon^{2t}\,\mathcal{R}_{in}^t}\right)\right\}\,.\end{aligned}$$

Using substitution of the expression (3.8) for n_{in}, the latter equation can be rewritten as

$$\begin{aligned}\mathsf{Prob}_e(\mathbb{C}_{cont}) \;&<\; \exp\Bigg\{ -\frac{N_{cont}\vartheta'\varepsilon\cdot\varepsilon^2\,\mathcal{R}_{in}}{2\varrho\ (\log_2(1/\varepsilon)+\Theta(1))}\quad\cdot \\ &\qquad \Big((2t-1)\ln(1/\varepsilon) + t\ln(1/\mathcal{R}_{in}) + \ t\ln\ln(1/\varepsilon) + \Theta(1)\Big)\Bigg\}\,. \qquad (3.11)\end{aligned}$$

The dominating term in the expression

$$(2t-1)\ln(1/\varepsilon) + t\ln(1/\mathcal{R}_{in}) + t\ln\ln(1/\varepsilon) + \Theta(1)$$

is $(2t-1)\ln(1/\varepsilon)$. By taking into account that $\mathcal{R}_{in} = \mathsf{C}(1-O(\varepsilon))$, Equation (3.11) can be rewritten, when ignoring all but the main term, as

$$\mathsf{Prob}_e(\mathbb{C}_{cont}) < \exp\left\{ -N_{cont}\cdot\left(\frac{(2t-1)\,\vartheta'\,\varepsilon^3\,\mathsf{C}}{2\varrho\cdot\log_2\mathsf{e}} + o(\varepsilon^3)\right)\right\}\,.$$

Thus, the decoding error probability is given by

$$\mathsf{Prob}_e(\mathbb{C}_{cont}) < \exp\{-N_{cont}\cdot\mathbb{E}(\mathsf{C},\varepsilon)\}\ ,$$

where

$$\begin{aligned}\mathbb{E}(\mathsf{C},\varepsilon) &= \max_{\varrho,\vartheta'}\left\{\frac{\vartheta'}{\varrho}\right\}\cdot\frac{(2t-1)\,\mathsf{C}}{2\cdot\log_2\mathsf{e}}\cdot\varepsilon^3\\ &= \max_{\kappa,\,\eta,\,\varrho}\left\{\frac{\eta(1-\kappa)}{2\varrho}-2\sqrt{\frac{\eta}{\varrho^3(1-\eta)}}\right\}\cdot\frac{(2t-1)\,\mathsf{C}}{2\cdot\log_2\mathsf{e}}\cdot\varepsilon^3\ ,\end{aligned}\tag{3.12}$$

and the parameters (κ,η,ϱ) are taken over

$$\kappa\in(0,1)\ ;\ \eta\in(0,1)\ ;\ \varrho>\frac{16}{\eta(1-\eta)(1-\kappa)^2}\ .\tag{3.13}$$

Next, we optimize the value of the constant

$$\Upsilon=\max_{\kappa,\,\eta,\,\varrho}\left\{\frac{\eta(1-\kappa)}{2\varrho}-2\sqrt{\frac{\eta}{\varrho^3(1-\eta)}}\right\}\ .$$

It is easy to see that the maximum is received for $\kappa\to0$. We substitute $\kappa=0$ in the expression (3.12) to obtain

$$\Upsilon=\max_{\eta,\,\varrho}\left\{\frac{\eta}{2\varrho}-2\sqrt{\frac{\eta}{\varrho^3(1-\eta)}}\right\}\ .\tag{3.14}$$

By taking a derivative of Υ over ϱ and comparing it to zero, we obtain that

$$\varrho=\frac{36}{\eta(1-\eta)}\ .$$

By substituting it back to the expression (3.14) and finding its maximum, we have $\eta=2/3$ and $\varrho=162$. These values obviously satisfy condition (3.13). The appropriate value of Υ is then

$$\begin{aligned}\Upsilon &= \frac{\eta}{2\varrho}-2\sqrt{\frac{\eta}{\varrho^3(1-\eta)}}=\frac{2/3}{2\cdot162}-2\sqrt{\frac{2/3}{162^3\cdot(1/3)}}\\ &= \frac{1}{1458}=6.8587\cdot10^{-4}\ .\end{aligned}$$

Finally, we have

$$\mathbb{E}(\mathsf{C},\varepsilon)=\frac{(2t-1)\,\mathsf{C}}{2916\cdot\log_2\mathsf{e}}\cdot\varepsilon^3\ .$$

Figure 3.1 shows the value of the error exponent $\mathbb{E}(\mathsf{C},\varepsilon)$ in the example for $t=1$, 2 and 3.
□

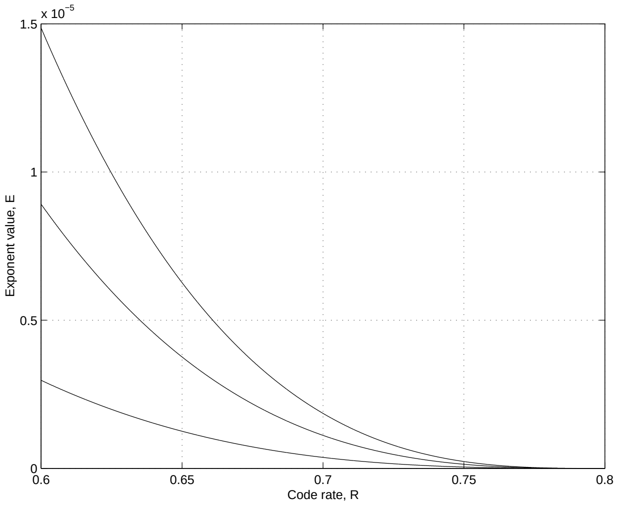

Figure 3.1: Error exponent $\mathbb{E}(\mathsf{C}, \varepsilon)$ for the code $\mathbb{C}_{cont}$.

Selection: $\mathsf{Prob}_e(\mathcal{C}_{in}) = 1/n_{in}^t$; $\mathsf{C} = 0.8$; $t = 1, 2, 3$ (bottom to top).

3.3.3 Decoding complexity

In this section, we show that under the assumption in Section 3.2.1 on the decoding time complexity of the code $\mathcal{C}_{in}$, and if the parameters of the codes are selected as in the proof of Theorem 3.3.2, then the decoding time complexity of the respective code $\mathbb{C}_{cont}$ is linear in the overall length N_{cont} and inverse polynomial in the gap ε from capacity.

Theorem 3.3.3 *Consider a BSC with crossover probability p and capacity $\mathsf{C} = \mathsf{C}_2(p)$. Let $\mathcal{R} = (1 - \varepsilon)\mathsf{C}$ be a design rate. Suppose that the following two conditions hold:*

(i) Let $\mathbb{C}_\Phi$ be a (family of) code defined in Section 3.2.2 of rate $R_\Phi = (1 - \varepsilon)/(1 - \kappa\varepsilon)$, $\kappa \in (0, 1)$ is a constant, over the smallest alphabet Φ satisfying $\log_2 |\Phi| \geq 1/(\kappa\varepsilon)^{h_0}$ from a dense sequence $\{\log_2 |\Phi_i|\}_{i=1}^{\infty}$, and $h_0 > 0$ is a constant.

(ii) Let $\mathcal{C}_{in}$ be a code of rate $\mathcal{R}_{in} = (1 - \kappa\varepsilon)\mathsf{C}$ with a decoding complexity over a BSC of capacity C given by

$$O\left(n_{in}^{\mathsf{s}} \cdot \frac{1}{\varepsilon^{\mathsf{r}}}\right) ,$$

where $\mathsf{s}, \mathsf{r} \geq 1$ are some constants.

Then, the time complexity of the respective code $\mathbb{C}_{cont}$, when decoded by $\mathcal{D}_{cont}$, is given by

$$N_{cont} \cdot \textsc{Poly}(1/\varepsilon) .$$

Proof. Below we count the total number of operations when decoding the code $\mathbb{C}_{cont}$ by the decoder $\mathcal{D}_{cont}$. There are two main steps.

- Step 1: n applications of the decoder $\mathcal{D}_{in}$ to the binary word of length n_{in}.
- Step 2: one application of the decoder $\mathcal{D}_{\Phi}$ to the word of length n over Φ.

In addition, there are n applications of each of the mappings $\mathcal{E}_0$ and $\mathcal{E}_0^{-1}$.

We separately count the number of operations during each step.

- Step 1: By the assumption on the decoding complexity of $\mathcal{D}_{in}$, n applications of this decoder result in time

$$O\left(n \cdot n_{in}^{\mathsf{s}} \cdot \frac{1}{\varepsilon^{\mathsf{r}}}\right) = O\left(N_{cont} \cdot n_{in}^{\mathsf{s}-1} \cdot \frac{1}{\varepsilon^{\mathsf{r}}}\right) . \tag{3.15}$$

From the definition of $\mathbb{C}_{cont}$, $n_{in} = \log_2 |\Phi| \,/\, \mathcal{R}_{in}$, so, we have

$$n_{in} = \frac{\log_2 |\Phi|}{(1-\kappa\varepsilon)\mathsf{C}} .$$

By using the density of values of $\log_2 |\Phi|$, we have $\log_2 |\Phi| \in \textsc{Poly}(1/\varepsilon)$, thus yielding $n_{in} \in \textsc{Poly}(1/\varepsilon)$. By substitution into (3.15), we obtain that the time complexity of Step 1 is $N_{cont} \cdot \textsc{Poly}(1/\varepsilon)$.

- Step 2: it is shown in Lemma B.1 that the number of applications of the decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ on the word of $\mathbb{C}_\Phi$ of length n over Φ is bounded by $\omega \cdot n$, where

$$\omega = 2 \cdot \left\lceil \frac{\ln\left(\dfrac{\Delta\beta\sqrt{\sigma}}{\beta-\sigma}\right)}{\ln\left(\dfrac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)} \right\rceil + \frac{1+\dfrac{\delta_A}{\delta_B}}{1-\left(\dfrac{4\gamma_{\mathcal{G}}^2}{\delta_A\delta_B}\right)^2} ,$$

and σ is an actual number of errors in the word. Thus, if the ratio σ/β is bounded away from 1, and $\mathcal{G}$ is a Ramanujan graph, then the value of ω is bounded from above by an absolute constant (independent of Δ).

The decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ are applied to words of length $\Delta \in \textsc{Poly}(1/\varepsilon)$. When classical decoders for GRS codes are used (such as Berlekamp-Massey), their complexity is polynomial in $1/\varepsilon$. Therefore, the decoding complexity in Step 2 is bounded by

$$n \cdot \textsc{Poly}(1/\varepsilon) \leq N_{cont} \cdot \textsc{Poly}(1/\varepsilon) .$$

Each application of the mapping $\mathcal{E}_0$ or $\mathcal{E}_0^{-1}$ is equivalent to multiplication of a vector by a matrix, where the number of rows and columns in the matrix is $\textsc{Poly}(1/\varepsilon)$. This can be done in time $\textsc{Poly}(1/\varepsilon)$.

Summing up the decoding complexities of all steps of the decoder, we obtain that the total number of operations is bounded by

$$N_{cont} \cdot \textsc{Poly}(1/\varepsilon) \; .$$

□

Note. The result in Theorem 3.3.3 is still valid if the outer code $\mathbb{C}_\Phi$ is replaced by any other code of rate $1 - \Theta(\varepsilon)$, whose decoding time complexity is linear in n and polynomial in $1/\varepsilon$, for a sequence of alphabets $\{\Phi\}_{i=1}^{\infty}$ such that $\{\log_2 |\Phi_i|\}_{i=1}^{\infty}$ is a dense sequence.

3.4 Time complexity of decoder in [10]

Similarly to Section 3.3, assume in this and the next sections that C is the capacity of the BSC with crossover probability p, and the design code rate is $\mathcal{R} = (1-\varepsilon)\mathsf{C}$. Our purpose is to compare the parameters of the codes from Section 3.3 with the codes presented by Barg and Zémor in [10] and [8] (with their respective decoding algorithms). In the sequel we show that the parameters of the codes from [10] and [8] cannot be modified such that the decoding time complexity is only sub-exponential in $1/\varepsilon$ while keeping a non-zero error exponent. The reason is this: both decoding algorithms in [10] and [8] make use of sub-routines (decoders for small constituent codes) that have time complexity exponential in the degree of underlying expander graph. This degree, in turn, depends (at least) polynomially on $1/\varepsilon$.

3.4.1 Analysis: binary codes

Consider the codes $\mathbb{C}_{BZ2}$ in [10] with their corresponding decoder, as defined in Section 1.4.3. The analysis of the codes $\mathbb{C}_{BZ2}$ is divided into two cases. In this section, we consider the case where the codes $\mathcal{C}_A$ and $\mathcal{C}_B$ are over $\mathbb{F} = \mathrm{GF}(2)$. In the next section, the analysis is generalized toward alphabet sizes which are large powers of 2.

In the binary case, following the analysis of [10], it is possible to show that for the code $\mathbb{C}_{BZ2}$ with the decoder $\mathcal{D}_{BZ2}$, the decoding error probability, $\mathsf{Prob}_e(\mathbb{C}_{BZ2})$, is bounded by

$$\mathsf{Prob}_e(\mathbb{C}_{BZ2}, p) \leq \exp\{-\alpha N f_3(\mathcal{R}, p)\} \; ,$$

where $0 < \alpha < 1$, and the main term of $f_3(\mathcal{R}, p)$ is less than or equal to

$$\max_{\mathcal{R} \leq R_0 < \mathsf{C}} \left\{ \mathbb{E}_0(R_0, p) \left(\tfrac{\mathsf{H}_2^{-1}(R_0 - \mathcal{R})}{2} - \Theta\left(\tfrac{1}{\sqrt{\Delta}}\right) \right) \right\} , \qquad (3.16)$$

and $\mathbb{E}_0(R_0, p)$ is the *random coding exponent* for rate R_0 over the BSC with crossover probability p.

Proposition 3.4.1 *If the codes $\mathbb{C}_{BZ2}$ (binary, as assumed in this section), have a positive error exponent under the decoding by $\mathcal{D}_{BZ2}$, then $\Delta = \Omega\left(1/(\mathsf{H}_2^{-1}(\varepsilon))^2\right)$.*

Proof. In order to have a positive error exponent it is needed that

$$\frac{\mathsf{H}_2^{-1}(R_0 - \mathcal{R})}{2} - \Theta\left(\frac{1}{\sqrt{\Delta}}\right) > 0 \ .$$

Observe that $R_0 - \mathcal{R} \le \mathsf{C} - \mathcal{R} = \mathsf{C}\varepsilon \le \varepsilon$. It follows from (3.16) that

$$\tfrac{1}{2}\mathsf{H}_2^{-1}(\varepsilon) \ge \tfrac{1}{2}\mathsf{H}_2^{-1}(R_0 - \mathcal{R}) > \Theta\left(1/\sqrt{\Delta}\right) ,$$

and thus $\Delta = \Omega\left(1/(\mathsf{H}_2^{-1}(\varepsilon))^2\right)$. □

It is suggested in [10] to use maximum-likelihood decoding for the codes $\mathcal{C}_A$ and $\mathcal{C}_B$. The known maximum-likelihood decoding algorithms, however, have per-bit time complexity at least

$$\exp\{\Omega(\Delta)\} = \exp\{\Omega\left(1/(\mathsf{H}_2^{-1}(\varepsilon))^2\right)\} \ .$$

3.4.2 Analysis: codes over large alphabets

Suppose that $\mathbb{F} = (\mathrm{GF}(2))^\ell$, $\ell \in \mathbb{N}$, and the codes $\mathcal{C}_A$ and $\mathcal{C}_B$ are chosen to satisfy properties (a)-(c) of Case 2 in Section 1.4.3. Then, for the code $\mathbb{C}_{BZ2}$ under the decoding by $\mathcal{D}_{BZ2}$, the decoding error probability $\mathsf{Prob}_e(\mathbb{C}_{BZ2})$ is bounded by

$$\mathsf{Prob}_e(\mathbb{C}_{BZ2}, p) \le \exp\{-\alpha N f_2(\mathcal{R}, p)\} \ ,$$

and the main term of $f_2(\mathcal{R}, p)$ is less than or equal to

$$\max_{\mathcal{R} \le R_0 < \mathsf{C}} \left\{ \mathbb{E}_0(R_0, p) \left(\tfrac{R_0 - \mathcal{R}}{2} - \Theta\left(\tfrac{1}{\sqrt{\Delta}}\right)\right)\right\} \ .$$

In this case, Proposition 3.4.1 can be rewritten as:

Proposition 3.4.2 *If the codes $\mathbb{C}_{BZ2}$ (over large $\mathbb{F}$, as assumed in this section) have a positive error exponent under the decoding by $\mathcal{D}_{BZ2}$, then $\Delta = \Omega\left(1/\varepsilon^2\right)$.*

The proof is very similar to that of Proposition 3.4.1.

When using known maximum-likelihood decoders for the codes $\mathcal{C}_A$ and $\mathcal{C}_B$, the decoding time complexity is at least

$$\exp\{\Omega(\Delta)\} = \exp\{\Omega\left(1/\varepsilon^2\right)\} \ .$$

3.5 Time complexity of decoder in [8]

Consider the codes $\mathbb{C}_{BZ3}$ in [8] with their corresponding decoder, as defined in Section 1.4.4. In the sequel, we analyze the decoding time complexity of these codes.

Lemma 3.5.1 *Let p satisfy $0 < p < \frac{1}{2}$, and let $0 < \varepsilon \ll p$. Then,*

$$\mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \varepsilon(1 - \mathsf{H}_2(p))\right) = p + \frac{\varepsilon(1 - \mathsf{H}_2(p))}{\log_2\left((1-p)/p\right)} - \frac{\varepsilon^2(1 - \mathsf{H}_2(p))^2 \log_2 \mathsf{e}}{2p(p-1)\left(\log_2\left((1-p)/p\right)\right)^3} + O(\varepsilon^3).$$

The proof of this lemma appears in Appendix D.

Proposition 3.5.2 *Let $\mathsf{C} = \mathsf{C}_2(p)$ be capacity of a BSC with crossover probability p. The decoding error probability of a random code of rate $\mathcal{R} = (1-\varepsilon)\mathsf{C}$, under maximum-likelihood decoding, behaves as $\exp\{-\Theta(\varepsilon^2)\}$ when $\varepsilon \to 0$.*

Proof. We start with the well-known expression for the probability exponent of the decoding error of a random code under maximum-likelihood decoding [34], [35]:

$$\mathbb{E}_0(\mathcal{R}, p) = \begin{cases} T(\delta, p) + \mathcal{R} - 1 & \text{if } \mathcal{R}_{crit} \le \mathcal{R} < \mathsf{C} \\ 1 - \log_2\left(1 + \sqrt{4p(1-p)}\right) - \mathcal{R} & \text{if } \mathcal{R}_{min} \le \mathcal{R} < \mathcal{R}_{crit} \\ -\delta \log_2 \sqrt{4p(1-p)} & \text{if } 0 \le \mathcal{R} < \mathcal{R}_{min}\ , \end{cases}$$

where $\mathcal{R}_{min}$ and $\mathcal{R}_{crit}$ are some threshold rates,

$$\delta = \delta_{GV}(\mathcal{R}) = \mathsf{H}_2^{-1}(1 - \mathcal{R})\ ,$$

and

$$T(x, y) = -x \log_2 y - (1-x)\log_2(1-y)\ .$$

At code rates $\mathcal{R}$ which are close to C, the relevant expression for the random coding exponent becomes

$$\mathbb{E}_0(\mathcal{R}, p) = T(\delta, p) + \mathcal{R} - 1\ . \tag{3.17}$$

Next, we express all terms of the relevant part of (3.17) in terms of ε. We recall that $\mathcal{R} = (1-\varepsilon)(1 - \mathsf{H}_2(p))$ and, thus,

$$\mathsf{H}_2^{-1}(1 - \mathcal{R}) = \mathsf{H}_2^{-1}(\varepsilon + \mathsf{H}_2(p) - \varepsilon\mathsf{H}_2(p))\ .$$

When disregarding the $O(\varepsilon^3)$ term, Equation (3.17) becomes

$$\begin{aligned}
\mathbb{E}_0(\mathcal{R},p) &= (1-\varepsilon)(1-\mathsf{H}_2(p)) - 1 + T\left(\mathsf{H}_2^{-1}(\varepsilon+\mathsf{H}_2(p)-\varepsilon\mathsf{H}_2(p)),\ p\right) \\
&\overset{(*)}{=} -\varepsilon-(1-\varepsilon)\mathsf{H}_2(p) + T\Bigg(p+\frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} \\
&\qquad\qquad - \frac{\varepsilon^2(1-\mathsf{H}_2(p))^2\log_2\mathsf{e}}{2p(p-1)\left(\log_2((1-p)/p)\right)^3},\ p\Bigg) \\
&= -\varepsilon-(1-\varepsilon)\mathsf{H}_2(p) - \Bigg(p+\frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} \\
&\qquad -\frac{\varepsilon^2(1-\mathsf{H}_2(p))^2\log_2\mathsf{e}}{2p(p-1)\left(\log_2((1-p)/p)\right)^3}\Bigg)\log_2 p \\
&\qquad -\Bigg(1-p-\frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} \\
&\qquad +\frac{\varepsilon^2(1-\mathsf{H}_2(p))^2\log_2\mathsf{e}}{2p(p-1)\left(\log_2((1-p)/p)\right)^3}\Bigg)\log_2(1-p) \\
&= -\varepsilon(1-\mathsf{H}_2(p)) + \frac{\varepsilon(1-\mathsf{H}_2(p))(-\log_2 p+\log_2(1-p))}{\log_2((1-p)/p)} \\
&\qquad +\frac{\varepsilon^2(1-\mathsf{H}_2(p))^2\log_2\mathsf{e}(\log_2 p-\log_2(1-p))}{2p(p-1)\left(\log_2((1-p)/p)\right)^3} \\
&= \frac{\varepsilon^2(1-\mathsf{H}_2(p))^2\log_2\mathsf{e}}{2p(1-p)\left(\log_2((1-p)/p)\right)^2} = \varepsilon^2\cdot c_p\ ,
\end{aligned}$$

where $c_p > 0$ is a constant that depends only on the crossover probability p of the channel. Note that the transition $(*)$ follows from Lemma 3.5.1. □

Proposition 3.5.3 *If the codes $\mathbb{C}_{BZ3}$ have a positive error exponent, then $\Delta = \Omega(1/\varepsilon^2)$.*

Proof. It is shown in [8] that the decoding error probability of the code $\mathbb{C}_{BZ3}$, $\mathsf{Prob}_e(\mathbb{C}_{BZ3})$, satisfies

$$\mathsf{Prob}_e(\mathbb{C}_{BZ3}) \le \exp\left\{-n\Delta l\delta_1(1+\alpha)^{-1}\cdot(\mathbb{E}_0(R_0,p)-M\alpha)(1-o(1))\right\},$$

where α is a constant defined in [8] (in paritcular, $1>\alpha>2\lambda_1/d_1$), and

$$M = M(\mathcal{R},p) = \begin{cases} \frac{1}{2}\log_2((1-p)/p) & \text{if } \mathcal{R}\le\mathcal{R}_{crit} \\ \log_2\left(\frac{\delta_{GV}(\mathcal{R})(1-p)}{(1-\delta_{GV}(\mathcal{R}))p}\right) & \text{if } \mathcal{R}\ge\mathcal{R}_{crit}\end{cases}\ ;$$

here $\delta_{GV}(\mathcal{R}) = \mathsf{H}_2^{-1}(1-\mathcal{R})$ is the Gilbert-Varshamov relative minimum distance for the rate $\mathcal{R}$, and $\mathcal{R}_{crit}$ is a threshold rate (see [35] for details).

We are interested in small values of ε, i.e. $\mathcal{R} \geq \mathcal{R}_{crit}$. In this case, the value of $M(\mathcal{R}, p)$ can be rewritten as

$$\begin{aligned} M(\mathcal{R}, p) &= \log_2 \left(\frac{\delta_{GV}(\mathcal{R})(1-p)}{(1-\delta_{GV}(\mathcal{R}))p} \right) \\ &= \log_2 \left(\frac{\mathsf{H}_2^{-1}(1-\mathcal{R})(1-p)}{(1-\mathsf{H}_2^{-1}(1-\mathcal{R}))p} \right) \\ &= \log_2 \left(\frac{\mathsf{H}_2^{-1}(\mathsf{H}_2(p)+\varepsilon-\varepsilon\mathsf{H}_2(p))(1-p)}{(1-\mathsf{H}_2^{-1}(\mathsf{H}_2(p)+\varepsilon-\varepsilon\mathsf{H}_2(p)))p} \right) , \end{aligned} \tag{3.18}$$

where the last transition is due to $\mathcal{R} = (1-\mathsf{H}_2(p))(1-\varepsilon)$. Using Lemma 3.5.1, the equality (3.18) becomes

$$M(\mathcal{R}, p) = \log_2 \frac{\left(p + \frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} - \frac{1}{2} \cdot \frac{\varepsilon^2(1-\mathsf{H}_2(p))^2 \log_2 \mathsf{e}}{p(p-1)(\log_2((1-p)/p))^3} \right)(1-p)}{\left(1 - p - \frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} + \frac{1}{2} \cdot \frac{\varepsilon^2(1-\mathsf{H}_2(p))^2 \log_2 \mathsf{e}}{p(p-1)(\log_2((1-p)/p)))^3} \right) p} + O(\varepsilon^3) \; .$$

When ignoring the terms of ε^2 and highest powers of ε, and denoting $\theta = \frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)}$, this equation becomes

$$\begin{aligned} M(\mathcal{R}, p) &= \log_2 \left(\frac{p+\theta}{1-p-\theta} \cdot \frac{1-p}{p} \right) + O(\theta^2) \\ &= \log_2 \left(\frac{1+\theta/p}{1-\theta/(1-p)} \right) + O(\theta^2) \\ &= \log_2 \left((1+\theta/p)(1+\theta/(1-p)) \right) + O(\theta^2) \\ &= \log_2 \left(1+\theta/p+\theta/(1-p) \right) + O(\theta^2) \; . \end{aligned}$$

Using Taylor's series for $\ln(\cdot)$ around 1 we obtain

$$\begin{aligned} M(\mathcal{R}, p) &= \log_2 \mathsf{e} \cdot \left(\frac{\theta}{p} + \frac{\theta}{(1-p)} \right) + O(\theta^2) \\ &= \frac{\log_2 \mathsf{e}}{p(1-p)} \cdot \theta + O(\theta^2) \; , \end{aligned}$$

and switching back to ε notation this becomes

$$M(\mathcal{R}, p) = \frac{\log_2 \mathsf{e}}{p(1-p)} \cdot \frac{\varepsilon(1-\mathsf{H}_2(p))}{\log_2((1-p)/p)} + O(\varepsilon^2) = \Theta(\varepsilon) \; . \tag{3.19}$$

Next, we estimate the value of α. Recall that $\alpha > 2\lambda_1/d_1$, and $d_1 \leq \Delta_1 \leq \Delta$. We have

$$\alpha > \frac{2\lambda_1}{d_1} \geq \frac{4\sqrt{\Delta_1 - 1}}{\Delta_1} \geq \frac{4\sqrt{\Delta - 1}}{\Delta} = \Theta\left(\frac{1}{\sqrt{\Delta}} \right) \; .$$

In order to have a positive error exponent it is necessary that

$$
\begin{aligned}
\mathbb{E}_0(R_0,p) - M\alpha > 0 \ &\Rightarrow \ \frac{\mathbb{E}_0(R_0,p)}{M} > \alpha \\
&\Rightarrow \ \frac{\mathbb{E}_0(R_0,p)}{M} > \Theta\left(\frac{1}{\sqrt{\Delta}}\right) \ .
\end{aligned}
$$

Using Proposition 3.5.2, $\mathbb{E}_0(R_0,p) = \Theta(\varepsilon^2)$, and thus from (3.19)

$$
\varepsilon = \Omega(1/\sqrt{\Delta}) \qquad \Rightarrow \qquad \Delta = \Omega(1/\varepsilon^2) \ .
$$

□

Assuming that the first two decoding iterations are as suggested in [8], we conclude that the time complexity of the decoding is $\exp\{\Omega(\Delta)\} = \exp\{\Omega(1/\varepsilon^2)\}$.

Chapter 4

Generalized Expander Codes

In this chapter, we generalize the codes defined in Chapter 2. The codes therein, for every vertex in the set A (or B), have the same set of constraints defined by the code $\mathcal{C}_A$ (or $\mathcal{C}_B$). By contrast, for the codes described in this chapter, there is more than one different set of constraints for the vertices in the set A (or B). We call such codes *generalized* expander codes. In the sequel, we analyze the parameters of these codes. Moreover, we present a linear-time decoding algorithm for the above codes. Finally, we show that the binary codes derived from the presented codes have a minimum distance at least as good as the minimum distance of the codes of Barg and Zémor [9], for a broad range of code rates.

4.1 Construction of generalized expander codes

Let $\mathcal{G} = (A : B, E)$ be a bipartite Δ-regular undirected connected graph with a vertex set $V = A \cup B$ such that $A \cap B = \emptyset$, and an edge set E such that there are no parallel edges in it, and every edge has one endpoint in A and one endpoint in B. We denote the size of A by n (clearly, n is also the size of B) and we will assume hereafter without any practical loss of generality that $n > 1$. We divide B into two sets, B^1 and B^2, such that $B^1 \cap B^2 = \emptyset$, $B^1 \cup B^2 = B$. Let $|B^2| = \eta n$, and thus $|B^1| = (1 - \eta)n$. The value $\eta \in [0, 1]$ will be defined in the sequel.

As before, for every vertex $u \in V$, we denote by $E(u)$ the set of edges that are incident with u. We assume an ordering on V, thereby inducing an ordering on the edges of $E(u)$ for every $u \in V$. For a word $\boldsymbol{z} = (z_e)_{e \in E}$ over an alphabet $\mathbb{F}$ we denote by $(\boldsymbol{z})_{E(u)}$ the sub-block of $\boldsymbol{z}$ that is indexed by $E(u)$. For a vertex $u \in V$ and a subset $S \subseteq V$ we denote by $\deg_S(u)$ the degree of u in the graph induced from $\mathcal{G}$ by the vertex set S.

Let $\mathbb{F}$ be the field $\mathrm{GF}(q)$ and let $\mathcal{C}_A$, $\mathcal{C}_1$ and $\mathcal{C}_2$ be linear $[\Delta, r_A\Delta, \delta_A\Delta]$, $[\Delta, r_1\Delta, \delta_1\Delta]$ and $[\Delta, r_2\Delta, \delta_2\Delta]$ codes over $\mathbb{F}$, respectively. Below, we generalize the code $\mathbb{C}$ presented in

Section 2.1. In Section 2.1, for any codeword $\boldsymbol{c} \in \mathbb{C}$, for any vertex $u \in B$, the vectors $(\boldsymbol{c})_{E(u)}$ belong to the same constituent code $\mathcal{C}_B$. In the current section, for any codeword $\boldsymbol{c} \in \mathbb{C}$, the sub-word $(\boldsymbol{c})_{E(u)}$ is in the code $\mathcal{C}_1$ if $u \in B^1$, and $(\boldsymbol{c})_{E(u)}$ is in $\mathcal{C}_2$ if $u \in B^2$, where $\mathcal{C}_1 \neq \mathcal{C}_2$.

More specifically, we define the code $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A, \mathcal{C}_1, \mathcal{C}_2)$ as the following linear code of length $|E|$ over $\mathbb{F}$:

$$\mathbb{C} = \Big\{ \boldsymbol{c} \in \mathbb{F}^{|E|} \; : \quad (\boldsymbol{c})_{E(u)} \in \mathcal{C}_A \text{ for every } u \in A, \;\; (\boldsymbol{c})_{E(u)} \in \mathcal{C}_1 \text{ for every } u \in B^1 \\ \text{and } (\boldsymbol{c})_{E(u)} \in \mathcal{C}_2 \text{ for every } u \in B^2 \Big\}$$

(for $\eta = 0, 1$ or, alternatively, for $\mathcal{C}_1 = \mathcal{C}_2$, the code $\mathbb{C}$ defined herein coincides with its counterpart defined in Section 2.1).

4.2 Bounds on the code parameters

Let $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A, \mathcal{C}_1, \mathcal{C}_2)$ and Φ be as defined in Section 4.1. We estimate the rate $\mathcal{R}$ of the code $\mathbb{C}$ by counting the number of parity-check equations. For each vertex in A, B^1 and B^2 there are $n\Delta(1-r_A)$, $n\Delta(1-\eta)(1-r_1)$ and $n\Delta\eta(1-r_2)$ parity-check equations, respectively. Some of these equations, however, may be linearly dependent. Summing these expressions, we obtain following bound on the rate of the code $\mathbb{C}$,

$$(1-\mathcal{R})n\Delta \leq n\Delta \left((1-r_A) + (1-\eta)(1-r_1) + \eta(1-r_2) \right) ,$$

which results in

$$\mathcal{R} \geq r_A + (1-\eta)r_1 + \eta r_2 - 1 \; . \tag{4.1}$$

Denote by $A_\mathcal{G}$ the adjacency matrix of $\mathcal{G}$. Recall that Δ is the largest eigenvalue of $A_\mathcal{G}$, and denote by $\gamma_\mathcal{G}$ the ratio between the second largest eigenvalue of $A_\mathcal{G}$ and Δ. Recall that when $\mathcal{G}$ is taken from a sequence of Ramanujan expander graphs with constant degree Δ, we have

$$\gamma_\mathcal{G} \leq \frac{2\sqrt{\Delta - 1}}{\Delta} \; .$$

Denote by $\mathcal{N}(u)$ the set of vertices in $\mathcal{G}$ adjacent to u.

We next turn to obtain several properties of the codewords of $\mathbb{C}$.

Proposition 4.2.1 *Suppose that $\delta_1 \geq \delta_2$. For any non-zero codeword $\boldsymbol{c}$ of $\mathbb{C}$ consider the following sequence $S_\boldsymbol{c}$ of sub-vectors of $\boldsymbol{c}$: $S_\boldsymbol{c} = \left((\boldsymbol{c})_{E(u)} \right)_{u \in B}$. Then the number of non-zero (vector) entries in $S_\boldsymbol{c}$ is at least*

$$n \cdot \frac{\delta_A - \gamma_\mathcal{G} \sqrt{\delta_A / \delta_2}}{1 - \gamma_\mathcal{G}} \; .$$

Proof. Define the code $\mathbb{C}_\Phi$ over Φ similarly to the definition in (2.4) with respect to the code $\mathbb{C}$ as defined in this chapter.

Consider a word $\boldsymbol{c}$ in the code $\mathbb{C}$. For every vertex $u \in B$, the sub-word $(\boldsymbol{c})_{E(u)}$ is a codeword of a code of a relative minimum distance $\geq \delta_2$. Then, for every vertex $u \in B$ such that $(\boldsymbol{c})_{E(u)}$ is not a zero vector, the number of non-zero entries in $(\boldsymbol{c})_{E(u)}$ is bounded from below by δ_2.

Theorem 2.2.1 deals with a case where for every vertex $u \in B$, the vectors $(\boldsymbol{c})_{E(u)}$ belong to the same code $\mathcal{C}_B$. However, the proof of that theorem uses the property that for every non-zero word $\boldsymbol{c} \in \mathbb{C}$, and for every vertex $u \in B$ such that $(\boldsymbol{c})_{E(u)}$ is not a zero vector, the number of non-zero entries in $(\boldsymbol{c})_{E(u)}$ is bounded from below by δ_B. This property is valid also in the case of the code $\mathbb{C}$ defined in the present section (with $\delta_2 = \delta_B$). Thus, all the steps of the proof of Theorem 2.2.1, with $\delta_B = \delta_2$ are valid also in the present case, and it follows that the relative minimum distance of the code $\mathbb{C}_\Phi$ over Φ is bounded from below by

$$\frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} .$$

This immediately yields the required conclusion. □

Proposition 4.2.2 *Let S be a subset of either A or B. Denote $\sigma = |S|/n$ and $J_S(u) = |S \cap \mathcal{N}(u)|$. Let $U = A$ if $S \subseteq B$ and $U = B$ if $u \subseteq A$. Then*

$$\frac{1}{n}\sum_{u \in U} (J_S(u) - \sigma\Delta)^2 \leq \gamma_{\mathcal{G}}^2 \Delta^2 \sigma(1-\sigma) . \tag{4.2}$$

Proof. The proof of this proposition appears as a guided exercise in [73, Problem 13.22].

Theorem 4.2.3 *Consider the code $\mathbb{C}$ over a field $\mathbb{F}$ with*

$$\eta < \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} - \gamma_{\mathcal{G}}^{2/3} . \tag{4.3}$$

Suppose that $\delta_2 \leq \delta_1$. Then, the relative minimum distance $\delta_{\mathbb{C}}$ of $\mathbb{C}$ satisfies

$$\delta_{\mathbb{C}} > \delta_A(\delta_1 - \tfrac{1}{2}\gamma_{\mathcal{G}}^{2/3}) . \tag{4.4}$$

Proof. First, it is easy to see that $\mathbb{C}$ is a linear subspace over $\mathbb{F}$, and thus the minimum distance of $\mathbb{C}$ equals the minimum weight of any non-zero codeword of $\mathbb{C}$.

Pick any non-zero codeword $\boldsymbol{c} \in \mathbb{C}$. Denote by $Y \subseteq E$ the support of $\boldsymbol{c}$, i.e.,

$$Y = \{e \in E : c_e \neq 0\} .$$

Let S and T be the sets of all vertices in A and B, respectively, that are endpoints of edges in Y. Let σ and τ denote ratios $|S|/n$ and $|T|/n$, respectively.

Next, we observe that using Proposition 4.2.2, the number ξ of vertices $u \in B$ for which

$$|J_S(u) - \sigma\Delta| \geq \rho\Delta\sqrt{\sigma(1-\sigma)} \ ,$$

for any $\rho \in (0,1)$, satisfies

$$\frac{\xi}{n}\left(\rho\Delta\sqrt{\sigma(1-\sigma)}\right)^2 \leq \gamma_{\mathcal{G}}^2\Delta^2\sigma(1-\sigma) \ .$$

It follows that

$$\frac{\xi}{n} \leq \frac{\gamma_{\mathcal{G}}^2}{\rho^2} \ . \tag{4.5}$$

Since for any $\sigma \in [0,1]$, $\sqrt{\sigma(1-\sigma)} \leq \frac{1}{2}$, we obtain that the number of vertices $u \in T$ for which

$$|J_S(u) - \sigma\Delta| \geq \frac{\rho\Delta}{2} \ , \tag{4.6}$$

is less than or equal to ξ (for any $\rho \in (0,1)$). Note that the next equation yields Equation (4.6):

$$J_S(u) \geq \sigma\Delta + \frac{\rho\Delta}{2} \ . \tag{4.7}$$

Pick $\rho = \gamma_{\mathcal{G}}^{2/3}$. We obtain that every vertex in T, except at most $\gamma_{\mathcal{G}}^{2/3}n$ vertices, has less than $\sigma\Delta + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta$ neighbors in S. Using Proposition 4.2.1, the relative size of T, τ, is at least

$$\tau \geq \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} \ . \tag{4.8}$$

There are at least $n(\tau - \eta - \gamma_{\mathcal{G}}^{2/3})$ vertices in $T \cap B^1$ that have less than $\sigma\Delta + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta$ neighbors in S. Using (4.3) and (4.8), there is at least one such vertex; denote it by v.

All edges e, such that $e \in Y$, have one endpoint in S. For the vertex v, the number of non-zero edges incident with it is at least $\delta_1\Delta$. On the other hand, the number $J_S(v)$ of neighbors (in S) of the vertex v is less than $\sigma\Delta + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta$. However, this number is greater than or equal to the number of the non-zero edges incident with v (note that there are no parallel edges in $\mathcal{G}$); thus,

$$\sigma\Delta + \tfrac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta > \delta_1\Delta \ ,$$

yielding that

$$\sigma > \delta_1 - \tfrac{1}{2}\gamma_{\mathcal{G}}^{2/3} \ . \tag{4.9}$$

The required result is obtained by the observation that each vertex in S has at least $\delta_A\Delta$ incident edges in Y. □

Next, we compare the code $\mathbb{C}$ with its counterpart presented in Chapter 2. It is shown in Theorems 4.2.3 and 2.2.1 that the relative minimum distances are given by approximately $\delta_A\delta_1$ and $\delta_A\delta_B$ for the code in this chapter and in Chapter 2, respectively, where δ_A, δ_1 and δ_B are the respective relative distances of the constituent codes. Take the constituent codes $\mathcal{C}_1$ and $\mathcal{C}_B$ to be equal. In such a case, the relative minimum distances of the code $\mathbb{C}$ and of its counterpart in Chapter 2 are almost equal (for sufficiently large values of Δ).

On the other hand, the rates of the code $\mathbb{C}$ and of its counterpart in Chapter 2, respectively, under the condition $\mathcal{C}_1 = \mathcal{C}_B$, are bounded from below by

$$r_A + (1-\eta)r_B + \eta r_2 - 1 ,$$

and

$$r_A + r_B - 1 ,$$

where $r_A, r_1 = r_B, r_2$ are the rates of the respective constituent codes (see Equalities (4.1) and (2.5)). When $0 < \eta < 1$ and $r_1 < r_2$, the rate of the code $\mathbb{C}$ defined in this section is strictly larger than the rate of its counterpart defined in Chapter 2. Therefore, the code in this chapter improves on the rate-distance ratio compared with the code $\mathbb{C}$ defined in Chapter 2.

Example 4.2.1 Consider the graph $\mathcal{G}$ and the code $\mathbb{C}$ defined in Section 4.1. Select some small $\epsilon > 0$, and let $\Delta \geq 1/\epsilon^3$ be an integer which is a feasible degree of a bipartite Ramanujan graph in [53], [62]. Then,

$$\gamma_{\mathcal{G}} = \frac{2\sqrt{\Delta-1}}{\Delta} < \frac{2}{\sqrt{\Delta}} \leq 2\epsilon^{3/2} .$$

Let $\mathbb{F} = \mathrm{GF}(q)$ such that $q = 2^\ell$ for some integer ℓ. We take the codes $\mathcal{C}_1$, $\mathcal{C}_2$ as GRS codes over $\mathbb{F}$ (and thus $\delta_1 + r_1 = \delta_2 + r_2 > 1$). We also take the code $\mathcal{C}_A$ as the code in Proposition 3 in [10], namely, $\mathcal{C}_A$ is a linear code over $\mathbb{F}$ which can be viewed as a linear binary code, with the following properties:

- The binary relative minimum distance, δ'_A, of $\mathcal{C}_A$ is bounded from below by

$$\delta'_A > \mathsf{H}_2^{-1}(1-r_A) - \varepsilon .$$

- The q-ary relative minimum distance, δ_A, of $\mathcal{C}_A$ is bounded from below by

$$\delta_A > 1 - r_A - \varepsilon ,$$

for sufficiently small value of ε.

Fix $\delta_2 = \epsilon$, and so $r_2 > 1 - \epsilon$. Let η be very close (but slightly less than) the value

$$\frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} - \gamma_{\mathcal{G}}^{2/3} = \frac{\delta_A - 2\epsilon^{3/2}\sqrt{\delta_A/\epsilon}}{1-2\epsilon^{3/2}} - 2^{2/3}\epsilon > \delta_A - 4\epsilon \ .$$

Then, using Equation (4.1), we have

$$\mathcal{R} \geq r_A + (1-\eta)r_1 + \eta r_2 - 1 > r_A + (1-\delta_A)r_1 + (\delta_A - 4\epsilon)(1-\epsilon) - 1 \ .$$

By taking $\epsilon \longrightarrow 0$ (and ignoring the vanishing terms), we obtain

$$\mathcal{R} \geq r_A + r_A r_1 + \delta_A - 1 = r_A r_1 \ .$$

Let $\boldsymbol{c} \in \mathbb{C}$ be a non-zero codeword, and let S and σ be defined as in the proof of Theorem 4.2.3 with respect to this $\boldsymbol{c}$. The conditions of Theorem 4.2.3 are satisfied in this case, and therefore, the equality (4.9) is implied. Then, the binary relative minimum distance of the code $\mathbb{C}$ is bounded from below by (when ignoring the vanishing terms)

$$\delta_{\mathbb{C}} > \left(\delta_1 - \tfrac{1}{2}\gamma_{\mathcal{G}}^{2/3}\right) \cdot \delta'_A = \left(\delta_1 - \frac{\epsilon}{2^{1/3}}\right) \cdot \delta'_A \xrightarrow{\epsilon \to 0} \delta_1 \delta'_A \geq (1-r_1)\mathsf{H}_2^{-1}(1-r_A) \ .$$

Next, suppose that some design rate $\mathcal{R}_d > 0$ is given. Let $\mathcal{C}_1$ be a code of rate r_1 over $\mathbb{F}$ that attains the Gilbert-Varshamov bound. By optimization over the rates of r_A and r_1, the lower bound on the binary relative minimum distance of $\mathbb{C}$ becomes arbitrarily close to

$$\max_{\mathcal{R}_d \leq r_A \leq 1} \left\{ \left(1 - \frac{\mathcal{R}_d}{r_A}\right) \mathsf{H}_2^{-1}\left(1 - r_A\right) \right\} \ ,$$

namely, the code $\mathbb{C}$ attains the Zyablov bound. □

4.3 Decoding

In this section we present the algorithm for decoding the generalized expander code $\mathbb{C}$ defined in Section 4.1. The decoding algorithm has a time complexity linear in n, and is able to correct a number of errors, which is almost half of the minimum distance given by the bound (4.4).

The structure of this section is as follows. First, we define a puncturing of the code $\mathbb{C}$ with respect to a subgraph of $\mathcal{G}$. We present an algorithm that is able to correct a significant number of errors for that punctured code. Next, we present an analysis of that algorithm, which is mainly based on the analysis in Section 2.3. Then, we show how the above algorithm can be used as a subroutine in order to correct many errors in the original code $\mathbb{C}$. Finally, we show the correctness of the decoding algorithm for the code $\mathbb{C}$.

4.3.1 Decoding of punctured expander code

In this section we consider a bipartite Δ-regular undirected connected graph $\mathcal{G} = (V, E)$ with the vertex set $V = A \cup B$ such that $A \cap B = \emptyset$, and an edge set E such that every edge in E has one endpoint in A and one endpoint in B. Let $|A| = |B| = n$. Let $\mathbb{F}$ be the field $\mathrm{GF}(q)$ and let $\gamma_{\mathcal{G}}$ be the ratio between the second largest eigenvalue of $A_{\mathcal{G}}$ and Δ. Let $\mathcal{C}(u)$ (for every $u \in V$) be a linear code of length Δ over $\mathbb{F}$ associated with the vertex u.

We define the code $\hat{\mathbb{C}} = \hat{\mathbb{C}}(\mathcal{G})$ as the following linear code of length $|E|$ over $\mathbb{F}$:

$$\hat{\mathbb{C}} = \left\{ \boldsymbol{c} \in \mathbb{F}^{|E|} \; : \quad (\boldsymbol{c})_{E(u)} \in \mathcal{C}(u) \text{ for every } u \in V \right\} .$$

Definition. Let $\tilde{\mathcal{G}} = (\tilde{V}, \tilde{E})$ be a subgraph of $\mathcal{G}$, $\tilde{V} \subseteq V$, and $\tilde{E} \subseteq E$ is an edge set induced from E by the vertex set $\tilde{V}$. The punctured code $\hat{\mathbb{C}}_p = \hat{\mathbb{C}}_p(\tilde{\mathcal{G}})$ with respect to the graph $\tilde{\mathcal{G}}$ is defined as

$$\hat{\mathbb{C}}_p(\tilde{\mathcal{G}}) = \left\{ \tilde{\boldsymbol{c}} \in \mathbb{F}^{|\tilde{E}|} : \exists \boldsymbol{c} \in \hat{\mathbb{C}} \text{ s.t. } \forall e \in \tilde{E} : \tilde{c}_e = c_e \right\} .$$

Note that the puncturing of $\hat{\mathbb{C}}$ induces puncturing of the codes $\mathcal{C}(u)$ for every $u \in \tilde{V}$. We denote the resulting punctured codes by $\mathcal{C}_p(u)$ for every $u \in \tilde{V}$. Suppose the existence of polynomial-time error-and-erasure decoders $\mathcal{D}(u)$ for the codes $\mathcal{C}(u)$, for every $u \in V$. Polynomial-time error-and-erasure decoders $\mathcal{D}_p(u)$ for the codes $\mathcal{C}_p(u)$ (for every $u \in \tilde{V}$) can be efficiently constructed from the decoders $\mathcal{D}(u)$.

Definition. Let $\tilde{\mathcal{G}} = (\tilde{V}, \tilde{E})$ be a subgraph of $\mathcal{G}$, such that $\tilde{V} \subseteq V$, and $\tilde{E} \subseteq E$ is an edge set induced from E by the vertex set $\tilde{V}$. Denote the corresponding punctured codes (as above) by $\mathcal{C}_p(u)$ (for $u \in \tilde{V}$). The *subgraph code* $\tilde{\mathbb{C}}_p = \tilde{\mathbb{C}}_p(\tilde{\mathcal{G}})$ with respect to the code $\hat{\mathbb{C}}$ and the graph $\mathcal{G}$ is defined as

$$\tilde{\mathbb{C}}_p(\tilde{\mathcal{G}}) = \left\{ \tilde{\boldsymbol{c}} \in \mathbb{F}^{|\tilde{E}|} : (\tilde{\boldsymbol{c}})_{\tilde{E}(u)} \in \mathcal{C}_p(u) \text{ for every } u \in \tilde{V} \right\} .$$

The following lemma follows immediately from the definitions of $\tilde{\mathbb{C}}_p$ and $\hat{\mathbb{C}}_p$.

Lemma 4.3.1 *The codes $\tilde{\mathbb{C}}_p$ and $\hat{\mathbb{C}}_p$ are related as follows:*

$$\hat{\mathbb{C}}_p \subseteq \tilde{\mathbb{C}}_p .$$

In this chapter, we denote by Φ the set of vectors of length Δ over $\mathbb{F} \cup \{?\}$. Define one-to-one linear mappings (for every $u \in \tilde{V} \cap A$) $\mathcal{E}_u : (\mathbb{F} \cup \{?\})^{\tilde{E}(u)} \to \Phi$ by

$$\mathcal{E}_u(\hat{\boldsymbol{a}}) = \boldsymbol{a} \quad \text{such that} \quad a_e = \begin{cases} \hat{a}_e & \text{for } e \in \tilde{E}(u) \\ 0 & \text{for } e \in E(u) \backslash \tilde{E}(u) \end{cases} .$$

Let the mapping $\psi_{\mathcal{E}} : \tilde{\mathbb{C}}_p \to \Phi^{|\tilde{V} \cap A|}$ be given by

$$\psi_{\mathcal{E}}(\boldsymbol{c}) = \Big(\mathcal{E}_u((\boldsymbol{c})_{\tilde{E}(u)}) \Big)_{u \in \tilde{V} \cap A} \ .$$

We define codes $\hat{\mathbb{C}}_\Phi$ and $\tilde{\mathbb{C}}_\Phi$ of length $|\tilde{V} \cap A|$ over Φ as follows.

$$\hat{\mathbb{C}}_\Phi = \Big\{ \psi_{\mathcal{E}}(\boldsymbol{c}) : \boldsymbol{c} \in \hat{\mathbb{C}}_p \Big\} \tag{4.10}$$

$$\tilde{\mathbb{C}}_\Phi = \Big\{ \psi_{\mathcal{E}}(\boldsymbol{c}) : \boldsymbol{c} \in \tilde{\mathbb{C}}_p \Big\} \tag{4.11}$$

The mapping $\psi_{\mathcal{E}}(\cdot)$ defines a correspondence between the codewords of the code $\tilde{\mathbb{C}}_p$ (or $\hat{\mathbb{C}}_p$) and the codewords of $\tilde{\mathbb{C}}_\Phi$ (or $\hat{\mathbb{C}}_\Phi$, respectively).

The following lemma follows from Lemma 4.3.1 and the definitions of the codes $\tilde{\mathbb{C}}_\Phi$ and $\hat{\mathbb{C}}_\Phi$.

Lemma 4.3.2 *The codes $\tilde{\mathbb{C}}_\Phi$ and $\hat{\mathbb{C}}_\Phi$ are related as follows:*

$$\hat{\mathbb{C}}_\Phi \subseteq \tilde{\mathbb{C}}_\Phi \ .$$

Definition. The puncturing of the code $\hat{\mathbb{C}}$ with respect to the graph $\tilde{\mathcal{G}}$ is called (d_A, d_B)-*preserving* if for every $u \in \tilde{V} \cap A$ and every $v \in \tilde{V} \cap B$, the codes $\mathcal{C}_p(u)$ and $\mathcal{C}_p(v)$ have minimum distance $\geq d_A$ and $\geq d_B$, respectively.

Lemma 4.3.3 *Suppose that the code $\hat{\mathbb{C}}$ is punctured (with respect to the graph $\tilde{\mathcal{G}}$) using (d_A, d_B)-preserving puncturing. Then, the minimum distance of each of the appropriate codes $\hat{\mathbb{C}}_\Phi$ and $\tilde{\mathbb{C}}_\Phi$ is greater than or equal to*

$$\frac{d_B/\Delta - \gamma_{\mathcal{G}} \sqrt{d_B/d_A}}{1 - \gamma_{\mathcal{G}}} \cdot n \ .$$

Proof. Consider a non-zero codeword of $\tilde{\mathbb{C}}_\Phi$ and let $\boldsymbol{c} \in \tilde{\mathbb{C}}_p$ be a corresponding codeword of $\tilde{\mathbb{C}}_p$. Let $\tilde{Y} \subseteq \tilde{E}$ be its support set. The set $\tilde{Y}$ is also a subset of E. Therefore, $\tilde{Y}$ forms a set of edges in the graph $\mathcal{G}$ such that for each vertex in A and B there are at least d_A and d_B edges in $\tilde{Y}$, respectively, incident with it.

Using the steps of the proof of Theorem 2.2.1, it follows that the number of vertices in A having incident edges in $\tilde{Y}$ is bounded from below by

$$\frac{d_B/\Delta - \gamma_{\mathcal{G}} \sqrt{d_B/d_A}}{1 - \gamma_{\mathcal{G}}} \cdot n \ ,$$

which is (due to linearity over $\mathbb{F}$ of the code $\tilde{\mathbb{C}}_\Phi$) also a lower bound on the minimum distance of the code $\tilde{\mathbb{C}}_\Phi$. From Lemma 4.3.2 it follows that this is also a lower bound on the minimum distance of the code $\hat{\mathbb{C}}_\Phi$. □

Figure 4.1 presents a decoder DecodeInduced for the code $\hat{\mathbb{C}}_\Phi$. This decoder is similar to the decoder in Figure 2.1.

Input: received word $\boldsymbol{y} = (\boldsymbol{y}_u)_{u \in \tilde{V} \cap A}$ in $(\Phi \cup \{?\})^{|\tilde{V} \cap A|}$.

Iteration 1: For $u \in A$ do: $\quad (\boldsymbol{z})_{\tilde{E}(u)} \leftarrow \begin{cases} \mathcal{E}_u^{-1}(\boldsymbol{y}_u) & \text{if } \exists \boldsymbol{a}_u \in \mathbb{F}^{\tilde{E}(u)} : \boldsymbol{y}_u = \mathcal{E}_u(\boldsymbol{a}_u) \\ ??\ldots? & \text{otherwise} \end{cases}$.

Iteration i: For $i = 2, 3, \ldots, \nu$ do:

(a) If i is odd then $U \equiv \tilde{V} \cap A$, else $U \equiv \tilde{V} \cap B$.

(b) For every $u \in U$ do: $(\boldsymbol{z})_{\tilde{E}(u)} \leftarrow$ application of $\mathcal{D}_p(u)$ on $(\boldsymbol{z})_{\tilde{E}(u)}$.

Output: $\psi_{\mathcal{E}}(\boldsymbol{z})$ if $\boldsymbol{z} \in \tilde{\mathbb{C}}_p$ (and declare 'error' otherwise).

Figure 4.1: Decoder $\mathsf{DecodeInduced}_{\tilde{\mathcal{G}}}$ for the induced code $\hat{\mathbb{C}}_\Phi$.

We use the notation "?" to stand for an erasure. The algorithm in Figure 4.1 makes use of a word $\boldsymbol{z} = (z_e)_{e \in \tilde{E}}$ over $\mathbb{F} \cup \{?\}$ that is initialized according to the contents of the received word $\boldsymbol{y}$ as follows. Each sub-block $(\boldsymbol{z})_{\tilde{E}(u)}$ that corresponds to a non-erased entry $\boldsymbol{y}_u$ of $\boldsymbol{y}$ such that there exists $\boldsymbol{a}_u \in \mathbb{F}^{\tilde{E}(u)}$ satisfying $\boldsymbol{y}_u = \mathcal{E}_u(\boldsymbol{a}_u)$, is initialized to the word $\boldsymbol{a}_u$. The remaining sub-blocks $(\boldsymbol{z})_{\tilde{E}(u)}$ are initialized as erased words of length $\deg_{\tilde{\mathcal{G}}}(u)$ (and thus it is possible that some erroneous symbols will be replaced by a block of erasures). Iterations $i = 3, 5, 7, \ldots$ use error-correcting decoders $\mathcal{D}_p(u) : \mathbb{F}^{\deg_{\tilde{\mathcal{G}}}(u)} \to \mathcal{C}_p(u)$ (for $u \in \tilde{V} \cap A$), and iterations $i = 2, 4, 6, \ldots$ use combined error-erasure decoders $\mathcal{D}_p(u) : \mathbb{F}^{\deg_{\tilde{\mathcal{G}}}(u)} \to \mathcal{C}_p(u)$ (for $u \in \tilde{V} \cap B$). The number of iterations ν is set to the number of iterations in the decoder in Figure 2.1, run with the same parameters, and is bounded by $O(\log n)$.

Pick a word $\boldsymbol{c} \in \hat{\mathbb{C}}_\Phi$. Suppose that some symbols of $\boldsymbol{c}$ are errors and some are erasures, resulting in the word $\boldsymbol{y} \in (\Phi \cup \{?\})^{|\tilde{V} \cap A|}$. Suppose that the decoder in Figure 4.1 is applied to the word $\boldsymbol{y}$.

We say that an edge $e \in \tilde{E}$ is *corrupted* at a given time during the execution of the algorithm (in Figure 4.1) if the respective entry in $\boldsymbol{z}$ is in $\mathbb{F}$ yet differs from $(\psi_{\mathcal{E}}^{-1}(\boldsymbol{c}))_e$. A vertex $v \in \tilde{V}$ is corrupted if $\tilde{E}(v)$ contains a corrupted edge.

For $i = 1, 2, \ldots, \nu$, denote by $\boldsymbol{z}^i = (z_e^i)_{e \in \tilde{E}}$ the result $\boldsymbol{z}$ at the end of Iteration i (value $i = 1$ is used for the word $\boldsymbol{z}$ just before the application of the decoder). We let Z_i be the respective subsets of corrupted edges, i.e.,

$$Z_i = \left\{ e \in \tilde{E} \; : \; z_e^i \in \mathbb{F} \text{ and } z_e^i \neq (\psi_{\mathcal{E}}^{-1}(\boldsymbol{c}))_e \right\} \; ,$$

and denote by S_i and R_i the respective subsets of corrupted and erased vertices, respectively:

$$\begin{aligned} S_i &= \left\{ v \in \tilde{V} \; : \; \tilde{E}(v) \cap Z_i \neq \emptyset \right\} \; , && (4.12) \\ R_i &= \left\{ v \in \tilde{V} \; : \; z_e = ? \;\text{ for every } e \in \tilde{E}(v) \right\} \; . && (4.13) \end{aligned}$$

(Actually, for $i \neq 1$, $R_i = \emptyset$).

Theorem 4.3.4 *Consider the (d_A, d_B)-preserving puncturing of the code $\hat{\mathbb{C}}$ with respect to the graph $\tilde{\mathcal{G}}$, and let*

$$\sqrt{d_A d_B} > 2\gamma_{\mathcal{G}} \Delta > 0 \; .$$

Then the decoder in Figure 4.1 applied to the word $\boldsymbol{y}$ (using the graph $\tilde{\mathcal{G}}$) recovers any pattern that consists of t errors (over Φ) and ρ erasures, provided that

$$t + \tfrac{1}{2}\rho < n \cdot \frac{(d_B/2\Delta) - \gamma_{\mathcal{G}} \sqrt{d_B/d_A}}{1 - \gamma_{\mathcal{G}}} \; .$$

The number of iterations is bounded from above by $O(\log n)$. The total time complexity of such decoding is $O(n)$.

Proof. The proof is similar to the proof of Theorem 2.3.1. For $i = 1, 2, \cdots, \nu$ define $\chi_i : V \to \{0, \frac{1}{2}, 1\}$ be the function

$$\chi_i(u) = \begin{cases} 1 & \text{if } u \in S_i \cap A \text{ and } i \text{ is odd} \\ 1 & \text{if } u \in S_i \cap B \text{ and } i \text{ is even} \\ \frac{1}{2} & \text{if } u \in R_1 \\ 0 & \text{otherwise} \end{cases} \; .$$

We denote

$$\sigma_i = \begin{cases} \dfrac{1}{n} \displaystyle\sum_{u \in \tilde{V} \cap A} \chi_i(u) & \text{if } i \text{ is odd} \\ \dfrac{1}{n} \displaystyle\sum_{u \in \tilde{V} \cap B} \chi_i(u) & \text{if } i \text{ is even} \end{cases} \; .$$

Note that a vertex $v \in \tilde{V} \cap A$ ($v \in \tilde{V} \cap B$, respectively) can belong to S_i (for $i \geq 2$) only if the sum $\sum_{u \in \mathcal{N}(v)} \chi_{i-1}(u)$ is at least $d_B/2$ ($d_A/2$, respectively). Then, the function χ_i satisfies the conditions of Lemma 2.3.2 (with d_A/Δ taken instead of δ_A for even i, and d_B/Δ instead of δ_B for odd i), and, so,

$$\sqrt{\frac{\sigma_{i-1}}{\sigma_i}} \geq \begin{cases} \dfrac{d_A/\Delta}{\gamma_{\mathcal{G}}} - \dfrac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\sigma_{i-1} & \text{for even } 0 < i < \ell \\ \dfrac{d_B/\Delta}{\gamma_{\mathcal{G}}} - \dfrac{1-\gamma_{\mathcal{G}}}{\gamma_{\mathcal{G}}}\sigma_{i-1} & \text{for odd } 0 < i < \ell \end{cases} . \tag{4.14}$$

The last equation coincides with equation (2.12). From this point, the proof continues as the proof of Theorem 2.3.1. This leads to the required result. □

4.3.2 General decoding procedure

In this section, we describe an error correcting procedure for the code $\mathbb{C}$, built using the graph $\mathcal{G}$ as in Section 4.1, with

$$0 < \eta < \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} - \gamma_{\mathcal{G}}^{2/3} .$$

First, we describe a construction procedure and parameters of the graph $\tilde{\mathcal{G}}$, which is built from the graph $\mathcal{G}$ and used in the decoding procedure. We define sets of vertices A', B' and $\tilde{V}$, as follows

$$A' = \left\{ v \in A \text{ s.t. } J_{B^2}(v) \geq \eta\Delta + \frac{\gamma_{\mathcal{G}}^{2/3}\Delta}{2} \right\} ,$$

$$B' = \left\{ u \in B^1 \text{ s.t. } |\mathcal{N}(u) \cap A'| \geq 2\gamma_{\mathcal{G}}^{2/3}\Delta \right\} .$$

and

$$\tilde{V} = V \backslash (A' \cup B') .$$

Let $\tilde{E}$ be the set

$$\tilde{E} = \left\{ e = \{u, v\} \text{ s.t. } u \in \tilde{V},\ v \in \tilde{V} \text{ and } e \in E \right\} .$$

Let $\tilde{\mathcal{G}}$ be a graph with the vertex set $\tilde{V}$ and the edge set $\tilde{E}$.

Lemma 4.3.5 *For every $u \in \tilde{V} \cap A$, the relative minimum distance of the code $\mathcal{C}_p(u)$ (corresponding to the puncturing defined by the graph $\tilde{\mathcal{G}}$ as above) is*

$$\delta_p(u) > \tfrac{1}{2}\gamma_{\mathcal{G}}^{2/3} .$$

Moreover, it holds that the size of A' satisfies $|A'| \leq \gamma_{\mathcal{G}}^{2/3} n$.

Proof. We make use of Proposition 4.2.2, with respect to the set B^2 instead of S. The number ξ of vertices $u \in A$ for which

$$|J_{B^2}(u) - \eta\Delta| \geq \rho\Delta\sqrt{\eta(1-\eta)}\ ,$$

for any $\rho \in (0,1)$ satisfies

$$\frac{\xi}{n}\left(\rho\Delta\sqrt{\eta(1-\eta)}\right)^2 \leq \gamma_{\mathcal{G}}^2\Delta^2\eta(1-\eta)\ ,$$

or, equivalently,

$$\xi \leq n \cdot \frac{\gamma_{\mathcal{G}}^2}{\rho^2}\ . \tag{4.15}$$

Since $\sqrt{\eta(1-\eta)} \leq \frac{1}{2}$, we conclude that this ξ is also an upper bound on the number of vertices $u \in A$ for which

$$|J_{B^2}(u) - \eta\Delta| \geq \frac{\rho\Delta}{2}\ . \tag{4.16}$$

We select $\rho = \gamma_{\mathcal{G}}^{2/3}$, and so $\xi \leq \gamma_{\mathcal{G}}^{2/3} n$. Then, the equation (4.16) is implied by

$$J_{B^2}(u) \geq \eta\Delta + \frac{\gamma_{\mathcal{G}}^{2/3}\Delta}{2}\ . \tag{4.17}$$

Therefore there are at most $\xi \leq \gamma_{\mathcal{G}}^{2/3} n$ vertices $u \in A$ that satisfy equation (4.17), and so $|A'| \leq \gamma_{\mathcal{G}}^{2/3} n$. This completes the proof of the second claim of the lemma.

Next, for every $u \in \tilde{V} \cap A$, there are less than $\eta\Delta + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta$ edges connecting u with vertices in B^2. Since the minimum distance of $\mathcal{C}(u)$ (for $u \in \tilde{V} \cap A$) is $\delta_A\Delta$, we obtain that the minimum relative distance of the code $\mathcal{C}_p(u)$ (for $u \in \tilde{V} \cap A$, and with respect to the graph $\tilde{\mathcal{G}}$) is bounded from below by

$$\begin{array}{rcl} \delta_p(u) & \geq & \delta_A - \left(\eta + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\right) \\ & > & \delta_A - \delta_A + \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2} + \gamma_{\mathcal{G}}^{2/3} - \frac{1}{2}\gamma_{\mathcal{G}}^{2/3} \\ & = & \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2} + \frac{1}{2}\gamma_{\mathcal{G}}^{2/3} \\ & > & \frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\ . \end{array}$$

□

Lemma 4.3.6 *For every $u \in \tilde{V} \cap B^1$, the relative minimum distance of the code $\mathcal{C}_p(u)$ is*

$$\delta_p(u) > \delta_1 - 2\gamma_{\mathcal{G}}^{2/3}\ ,$$

and the size of the set B' is bounded from above by

$$|B'| < \gamma_{\mathcal{G}}^{4/3} \cdot n\ .$$

Proof. We make use of Lemma 2.3.2. Denote by S the set A', and by T the set B' (in the notations of Lemma 2.3.2). Let $\sigma = |S|/n$ and $\tau = |T|/n$. For every $u \in S$ let $\chi(u) = 1$, for every $u \in T$ let $\chi(u) = 1$, and for every $u \in V \backslash (S \cup T)$ let $\chi(u) = 0$. Take the value of $\delta_B/2$ in Lemma 2.3.2 to be equal to $2\gamma_{\mathcal{G}}^{2/3}$. Then, we obtain that

$$\sqrt{\frac{\sigma}{\tau}} \geq \frac{2\gamma_{\mathcal{G}}^{2/3} - (1-\gamma_{\mathcal{G}})\sigma}{\gamma_{\mathcal{G}}} \; . \tag{4.18}$$

We substitute the inequality $\sigma \leq \gamma_{\mathcal{G}}^{2/3}$ (which is due to Lemma 4.3.5) into inequality (4.18), thus obtaining

$$\sqrt{\frac{\gamma_{\mathcal{G}}^{2/3}}{\tau}} \geq \frac{2\gamma_{\mathcal{G}}^{2/3} - (1-\gamma_{\mathcal{G}})\gamma_{\mathcal{G}}^{2/3}}{\gamma_{\mathcal{G}}} > \frac{\gamma_{\mathcal{G}}^{2/3}}{\gamma_{\mathcal{G}}} = \frac{1}{\gamma_{\mathcal{G}}^{1/3}} \; ,$$

which yields

$$|T| < \gamma_{\mathcal{G}}^{4/3} \cdot n \; .$$

For every $u \in \tilde{V} \cap B^1$, the number of edges in E connecting it with the vertices in A' is less than $2\gamma_{\mathcal{G}}^{2/3}\Delta$, and, therefore the relative minimum distance of $\mathcal{C}_p(u)$ is

$$\delta_p(u) > \delta_1 - 2\gamma_{\mathcal{G}}^{2/3} \; .$$

□

Corollary 4.3.7 *The subgraph $\tilde{\mathcal{G}}$ induces $\left(\frac{1}{2}\gamma_{\mathcal{G}}^{2/3}\Delta, (\delta_1 - 2\gamma_{\mathcal{G}}^{2/3})\Delta\right)$ - preserving puncturing on the code $\mathbb{C}$.*

Figure 4.2 below presents a decoder for the code $\mathbb{C}$. This decoder uses decoders $\mathsf{DecodeInduced}_{\tilde{\mathcal{G}}}$ in Figure 4.1 and $\mathsf{DecodeExpander}_{\mathcal{G}}$ in Figure 4.3 below as subroutines. The decoding consists of three phases. In Phase A, the decoder in Figure 4.2 restores all symbols which are indexed by the edges in the subgraph $\tilde{\mathcal{G}}$ (which are the symbols of the punctured code $\hat{\mathbb{C}}_p(\tilde{\mathcal{G}})$). In Phase B, the decoder in Figure 4.2 restores all the symbols of $\boldsymbol{z}$, which are indexed by the edges incident with the vertices in $B^1 \backslash B'$ (most of these edges were a part of the graph $\tilde{\mathcal{G}}$; however, some edges incident with the vertices in B^1 are connected to the vertices in A', so they were not recovered in Phase A). Finally, in Phase C, the decoder in Figure 4.2 marks all the symbols indexed by the edges incident with the vertices in B^2 as 'erasures', and restores their original values by an application of the decoder $\mathsf{DecodeExpander}_{\mathcal{G}}$ in Figure 4.3.

The decoder $\mathsf{DecodeExpander}_{\mathcal{G}}$ in Figure 4.3 is similar to the decoder in Figure 2.1. However, the decoder in Figure 4.3 is applied to the word $\boldsymbol{y} \in (\mathbb{F} \cup \{?\})^{|E|}$, in contrast to its counterpart in Figure 2.1, which is applied to a word in $(\Phi \cup \{?\})^n$. The initialization of $\boldsymbol{z}$ is replaced, and $\boldsymbol{z}$ is set to the input word $\boldsymbol{y}$ as is. The output of the decoder is the word $\boldsymbol{z}$ itself. The number of iterations ν' is set to the number of iterations in the decoder in Figure 2.1; in particular, ν' and is bounded from above by $O(\log n)$.

```
Input: Received word $\boldsymbol{y} = (\boldsymbol{y}_e)_{e \in E}$ in $\mathbb{F}^{|E|}$.

Phase A
    1. For every $v \in A$ let $(\boldsymbol{x})_{E(v)} \leftarrow$ application of $\mathcal{D}(v)$ on $(\boldsymbol{y})_{E(v)}$.
    2. For $h_t = 1$ to $\lceil \delta_A \Delta / 2 \rceil$ do {

        (i) For every $v \in A$ let
              $(\boldsymbol{z})_{E(v)} = \begin{cases} (\boldsymbol{x})_{E(v)} & \text{if } \mathsf{d}((\boldsymbol{y})_{E(v)}, (\boldsymbol{x})_{E(v)}) < h_t \\ ??? & \text{otherwise} \end{cases}$ .

        (ii) Let $\boldsymbol{w} \leftarrow \psi_{\mathcal{E}}((\boldsymbol{z})_{\tilde{E}(v)})$.

        (iii) Let $\boldsymbol{w} \leftarrow \mathsf{DecodeInduced}_{\tilde{\mathcal{G}}}(\boldsymbol{w})$.

        (iv) If ‘no error’ goto Phase B.

    }
    3. Return ‘error’.

Phase B
    1. $(\boldsymbol{z})_{\tilde{E}} \leftarrow \psi_{\mathcal{E}}^{-1}(\boldsymbol{w})$ .
    2. For every $v \in B^1 \backslash B'$ let $(\boldsymbol{z})_{E(v) \backslash \tilde{E}(v)} \leftarrow ???$ .
    3. For every $v \in B^1 \backslash B'$ let $(\boldsymbol{z})_{E(v)} \leftarrow$ application of $\mathcal{D}(v)$ on $(\boldsymbol{z})_{E(v)}$.

Phase C
    1. For every $v \in B^2 \cup B'$ let $(\boldsymbol{z})_{E(v)} \leftarrow ???$ .
    2. Let $\boldsymbol{z} \leftarrow \mathsf{DecodeExpander}_{\mathcal{G}}(\boldsymbol{z})$ .

Output: $\boldsymbol{z}$.
```

Figure 4.2: Decoder for the generalized expander code $\mathbb{C}$.

4.3.3 Decoding analysis

In this section we analyze the decoder in Figure 4.2. We show that if $\delta_1 > 2\gamma_{\mathcal{G}}^{2/3}$, then the decoder in Figure 4.2 is able to correct up to $\mathbb{J}_{\mathbb{C}}$ errors over $\mathbb{F}$ in the word $\boldsymbol{y}$, where $\mathbb{J}_{\mathbb{C}}$ is given by

$$\mathbb{J}_{\mathbb{C}} \triangleq \frac{\delta_1/2 - \gamma_{\mathcal{G}}^{2/3}\left(1 + \sqrt{2\left(\delta_1 - 2\gamma_{\mathcal{G}}^{2/3}\right)}\right)}{1 - \gamma_{\mathcal{G}}} \cdot \delta_A \Delta n \; . \tag{4.19}$$

Let $\boldsymbol{c} \in \mathbb{C}$ be the codeword such that $\mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) \leq \mathbb{J}_{\mathbb{C}}$.

The following proposition provides a relation between the number of corrupted vertices and the number of erased vertices in A after Step 2-(i) of Phase A of the decoder in Figure 4.2,

```
Input: received word $\boldsymbol{y} = (y_e)_{e \in E}$ in $(\mathbb{F} \cup \{?\})^{|E|}$.
Initialization: Let $\boldsymbol{z} \leftarrow \boldsymbol{y}$ .
Iteration $i$:   For $i = 2, \ldots, \nu'$ do:
    (a) If $i$ is odd then $U \equiv B$, else $U \equiv A$.
    (b) For every $u \in U$ do: $(\boldsymbol{z})_{E(u)} \leftarrow$ application of $\mathcal{D}(u)$ on $(\boldsymbol{z})_{E(u)}$.
Output: $\boldsymbol{z}$ if $\boldsymbol{z} \in \mathbb{C}$ (and declare 'error' otherwise).
```

Figure 4.3: Decoder $\mathsf{DecodeExpander}_{\mathcal{G}}$ for the code $\mathbb{C}$.

for some value of h_t. Following definitons (4.12) and (4.13), we define $S_1(h_t)$ and $R_1(h_t)$ to be the values of S_1 and R_1 in an application of the procedure DecodeInduced, while the external loop of Step 2 in Phase A assumes the value h_t of the threshold.

Proposition 4.3.8 *There exists a threshold*

$$h_t \in \{1, 2, \ldots, \lceil \delta_A \Delta / 2 \rceil\}$$

for which

$$2|S_1(h_t)| + |R_1(h_t)| \leq \frac{2\,\mathsf{d}(\boldsymbol{y}, \boldsymbol{c})}{\delta_A \Delta} \ .$$

The proof of Proposition 4.3.8 is similar to its GMD counterpart. The reader can refer to a proof of Theorem 2 in [31].

Proposition 4.3.9 *Let* $\boldsymbol{c} \in \mathbb{C}$. *Suppose that the decoder in Figure 4.2 is applied to the word* $\boldsymbol{y}$ *such that* $\mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) \leq \mathbb{J}_{\mathbb{C}}$. *Then Step 1 at Phase B ends with* $\boldsymbol{z}$ *such that*

$$\forall e \in \tilde{E} \ : \ z_e = c_e \ .$$

Proof. Let h_t be the threshold guaranteed by Proposition 4.3.8. For this h_t, the procedure DecodeInduced is applied to the word $\boldsymbol{w}$, having ϑ errors and ρ erasures, such that

$$\begin{aligned} \vartheta + \tfrac{1}{2}\rho &\leq |S_1(h_t)| + \tfrac{1}{2}|R_1(h_t)| \\ &\leq \frac{\delta_1/2 - \gamma_{\mathcal{G}}^{2/3}\left(1 + \sqrt{2\left(\delta_1 - 2\gamma_{\mathcal{G}}^{2/3}\right)}\right)}{1 - \gamma_{\mathcal{G}}} \cdot n \ . \end{aligned} \tag{4.20}$$

From Corollary 4.3.7 the puncturing of $\mathbb{C}$ (with respect to the graph $\tilde{\mathcal{G}}$) is a $(\gamma_{\mathcal{G}}^{2/3}\Delta/2, (\delta_1 - 2\gamma_{\mathcal{G}}^{2/3})\Delta)$-preserving puncturing, thereby satisfying the conditions of Theorem 4.3.4 (with $d_A = \gamma_{\mathcal{G}}^{2/3}\Delta/2$ and $d_B = (\delta_1 - 2\gamma_{\mathcal{G}}^{2/3})\Delta)$. Thus, the right-hand side of (4.20) is within the correction radius of the code $\tilde{\mathbb{C}}_\Phi$ under the decoder in Figure 4.1. Therefore, the word of the corresponding code $\tilde{\mathbb{C}}_\Phi$ will be restored by the procedure $\mathsf{DecodeInduced}$ when applied to $\boldsymbol{w}$. Consequently, by the definition of $\psi_{\mathcal{E}}$, at the end of Step 1 of Phase B,

$$\forall e \in \tilde{E} \;\; : \;\; z_e = c_e \; .$$

□

Proposition 4.3.10 *Let $\boldsymbol{c} \in \mathbb{C}$ and suppose that the decoder in Figure 4.2 is applied to the word $\boldsymbol{y}$ such that $\mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) \le \mathbb{J}_{\mathbb{C}}$. Then, at the end of Step 3 of Phase B, for every $u \in B^1 \backslash B'$,*

$$(\boldsymbol{z})_{E(u)} = (\boldsymbol{c})_{E(u)} \; .$$

Proof. Pick any $v \in B^1 \backslash B'$. It follows from Proposition 4.3.9 that for every $e \in \tilde{E}(v)$, $z_e = c_e$. From the definition of the graph $\tilde{\mathcal{G}}$, there are less than $2\gamma_{\mathcal{G}}^{2/3}\Delta$ edges $e \in E$ incident with v such that $e \notin \tilde{E}$. Given that $\delta_1 > 2\gamma_{\mathcal{G}}^{2/3}$, there is only one way to set the values of these $< 2\gamma_{\mathcal{G}}^{2/3}\Delta$ edges (which can be thought of as 'erasures') such that $(\boldsymbol{z})_{E(v)} \in \mathcal{C}(v)$. Thus, obviously, the application of the error-and-erasure decoder $\mathcal{D}(v)$ (for $v \in B^1$) in Step 3 of Phase B will result in $z_e = c_e$ for every $e \in E(v)$. □

Proposition 4.3.11 *Let $\boldsymbol{c} \in \mathbb{C}$ and suppose that the decoder in Figure 4.2 is applied to the word $\boldsymbol{y}$ such that $\mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) \le \mathbb{J}_{\mathbb{C}}$. Step 2 of Phase C of the decoder terminates with $\boldsymbol{z} = \boldsymbol{c}$.*

Proof. It follows from Proposition 4.3.10 that before the execution of Step 1 of Phase C, for every $u \in B^1 \backslash B'$ it holds that $(\boldsymbol{z})_{E(u)} = (\boldsymbol{c})_{E(u)}$. During the execution of Step 1 of Phase C, for every $u \in B^2 \cup B'$, $(\boldsymbol{z})_{E(u)}$ is initialized as a vector of 'erasures'. Therefore, before the application of procedure $\mathsf{DecodeExpander}$, there are $|B^2 \cup B'| < n(\eta + \gamma_{\mathcal{G}}^{4/3})$ erased vertices, and the rest of the vertices in B are correct (neither erroneous nor erased).

The code $\mathbb{C}$ has the property that for every $\boldsymbol{c} \in \mathbb{C}$, for every $u \in A$ (respectively, $u \in B$), $(\boldsymbol{c})_{E(u)}$ is a codeword of a code of relative distance δ_A (respectively, $\ge \delta_2$). Theorem 2.3.1 can be used with respect to the code $\mathbb{C}$ defined as in Section 2.1, where $\delta_B = \delta_2$, and all vertices in $B^2 \cup B'$ are considered as 'erased'. The decoding algorithm in Figure 4.3 is essentially the same as the decoding algorithm in Figure 2.1, starting with Iteration 2, when the sets A and B are switched; thus, the analysis as in Theorem 2.3.1 can be applied to it.

By using Theorem 2.3.1, the decoder in Figure 4.2 will restore the original word $\boldsymbol{c}$, given that the number of erased vertices in B is less than

$$n \cdot \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} \ .$$

Indeed, since $\gamma_{\mathcal{G}} < 1$, we have

$$n(\eta + \gamma_{\mathcal{G}}^{4/3}) < n \cdot \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} - \gamma_{\mathcal{G}}^{2/3} + \gamma_{\mathcal{G}}^{4/3} < n \cdot \frac{\delta_A - \gamma_{\mathcal{G}}\sqrt{\delta_A/\delta_2}}{1-\gamma_{\mathcal{G}}} \ ,$$

and thus the decoder in Figure 4.2 will halt with $\boldsymbol{z} = \boldsymbol{c}$. □

4.4 Distance properties of generalized expander codes

In this section, we consider the codes which are defined in Section 4.1. The aim of this section is to derive a lower bound on the binary codes induced by the codes therein.

Let $|\mathbb{F}| = q = 2^\ell$. Then, each symbol over $\mathbb{F}$ can be viewed as a vector of length ℓ over GF(2), and each codeword of $\mathbb{C}$ can be viewed as either over $\mathbb{F}$ or over GF(2). Denote by $\mathsf{w_b}(e)$ the relative binary weight of the symbol indexed by e, and by $\mathsf{w_b}(\boldsymbol{x})$ the relative binary weight of the word $\boldsymbol{x}$.

Pick some non-zero codeword $\boldsymbol{c} \in \mathbb{C}$. Denote by $Y \subseteq E$ the support of $\boldsymbol{c}$, i.e.,

$$Y = \{e \in E : c_e \neq 0\} \ .$$

Let S and T be the sets of all vertices in A and B, respectively, that are endpoints of edges in Y. Let σ and τ denote ratios $|S|/n$ and $|T|/n$, respectively. Denote

$$T^1 = T \cap B^1 \ ,$$

and let $\tau_1 = |T^1|/n$. It is shown in the proof of Theorem 4.2.3 that the value of τ_1 is bounded from below by a fixed constant (for any non-zero $\boldsymbol{c} \in \mathbb{C}$).

Let $\Gamma = \Gamma(\boldsymbol{c})$ be the average, over all edges $e \in E_{S \cup T^1}$ of the relative binary weight of c_e, i.e.

$$\Gamma = \frac{1}{|E_{S\cup T^1}|} \sum_{e \in E_{S\cup T^1}} \mathsf{w_b}(e) \ .$$

Let $v \in S \cup T^1$ be some vertex. We define local parameters B_v and Γ_v similarly to their definition in [9]. We recall the definitions below.

- The quantity B_v is defined as the average over all *non-zero* edges $e \in E_{S\cup T^1}$ incident with v, of the binary weight $\mathsf{w_b}(e)$.

- The quantity Γ_v is defined as the average, over all edges $e \in E_{S \cup T^1}$ incident with v, zero or not, of the binary weight $\mathsf{w_b}(e)$ of e.

For instance, if $v \in T^1$ then

$$\Gamma_v = \frac{1}{\deg_S(v)} \sum_{e \in E_{S \cup \{v\}}} \mathsf{w_b}(e) = \frac{1}{\deg_S(v)} \sum_{e \in E(v)} \mathsf{w_b}(e) \ ,$$

and, if $v \in S$ then

$$\Gamma_v = \frac{1}{\deg_{T^1}(v)} \sum_{e \in E_{T^1 \cup \{v\}}} \mathsf{w_b}(e) = \frac{1}{\deg_S(v)} \sum_{e \in E(v)} \mathsf{w_b}(e) \ .$$

If for some vertex $v \in S \cup T^1$ there are no incident edges in $E_{S \cup T^1}$, then Γ_v is set to zero. Similarly, if for some vertex $v \in S \cup T^1$ there are no non-zero incident edges in $E_{S \cup T^1}$, then B_v is set to zero.

Note that $\Gamma_v \leq \mathsf{B}_v$.

Along this section, similarly to the notation in [9], we denote by ε the value that can be made as small as desired by increasing Δ.

In this section, the proofs are similar to their counterparts in [9]. We will prove the lemmas that require some adjustments of their proofs, and will provide the rest of the claims without proofs.

Lemma 4.4.1 *(This lemma is a counterpart of Corollary 6 in [9].) Let α be such that $\alpha = o_\Delta(1)$ and $1/(\alpha\sqrt{\Delta}) = o_\Delta(1)$. Let $S \subseteq A$ such that $|S| = \sigma n$. Define*

$$R_\alpha = \{v \in B : (1-\alpha)\sigma\Delta \leq \deg_S(v) \leq (1+\alpha)\sigma\Delta\} \ . \tag{4.21}$$

Then, $1 - |R_\alpha|/n = o_\Delta(1)$.

This lemma directly follows from Proposition 4.2.2. It can also be shown by using the techniques presented in [9].

Lemma 4.4.2 *(This lemma is a counterpart of Lemma 7 in [9].) Using the notations defined above,*

$$\begin{aligned} \Gamma &= \frac{1}{|S|} \sum_{v \in S} \Gamma_v + o_\Delta(1) & (4.22) \\ &= \frac{1}{|T^1|} \sum_{v \in T^1} \Gamma_v + o_\Delta(1) \ . & (4.23) \end{aligned}$$

The proof of this lemma is similar to the proof of Lemma 7 in [9]. We reformulate and reprove the main stages of it in Appendix E.

Next, we define the q-ary constrained distance of the code $\mathcal{C}_B$, $\delta_B(\mathsf{B})$. It is a lower bound on the minimum relative q-ary weight of any non-zero codeword of $\mathcal{C}_1$ such that the average binary weight of its non-zero symbols equals $\mathsf{B}\ell$.

Lemma 4.4.3 *For any $\varepsilon > 0$, ℓ and Δ large enough, there exist codes $\mathcal{C}_B$ of rate r_B such that for any $0 < \mathsf{B} < 1$, the minimum relative B-constrained q-ary weight $\delta_B(\mathsf{B})$ of $\mathcal{C}_B$ satisfies*

$$\delta_B(\mathsf{B}) \geq \frac{1 - r_B}{\mathsf{H}_2(\mathsf{B})} - \varepsilon \; . \tag{4.24}$$

In particular, the code $\mathcal{C}_B$ can be selected as an appropriate RS code.

The claim of this lemma is not related to the definition of the code $\mathbb{C}$. This lemma appears as Lemma 9 in [9], and its proof appears therein as well.

Lemma 4.4.4 *For a non-zero codeword $\boldsymbol{c} \in \mathbb{C}$, let S, σ and Γ be defined as above. For any $\varepsilon > 0$ there exist Δ and ℓ such that for any $\boldsymbol{c} \in \mathbb{C}$:*

$$\sigma \geq \frac{1 - r_B}{\bar{\mathsf{H}}_2(\Gamma)} - \varepsilon \; , \tag{4.25}$$

where $\bar{\mathsf{H}}_2(\Gamma)$ is defined as

$$\bar{\mathsf{H}}_2(\Gamma) = \left\{ \begin{array}{cl} \mathsf{H}_2(\Gamma) & \textit{for } 0 \leq \Gamma \leq \frac{1}{2} \\ 1 & \textit{for } \frac{1}{2} < \Gamma \leq 1 \end{array} \right. \; . \tag{4.26}$$

This lemma is a counterpart of Lemma 10 in [9], and its proof is similar to the proof of Lemma 10 in [9]. In Appendix E, we reprove this lemma.

Let α be such that $\alpha = o_\Delta(1)$ and $1/\alpha\sqrt{\Delta} = o_\Delta(1)$. Define the set

$$S_\alpha = \{ v \in S : (1-\alpha)\tau_1\Delta \leq \deg_{T^1}(v) \leq (1+\alpha)\tau_1\Delta \} \; .$$

Then, by Lemma 4.4.1 (when switching between A and B, and between S and T_1), $1 - S_\alpha/n = o_\Delta(1)$.

Now we introduce the binary constrained distance $\delta_A(\mathsf{B})$ for the code $\mathcal{C}_A$. This definition is different from the definition of $\delta_B(\mathsf{B})$, in particular, because the constraint is applied only to the symbols indexed by the edges incident with B^1, and at the same time the symbols indexed by the edges incident with B^2 remain unconstrained. Let us fix a set of $(1-\alpha)\tau_1\Delta$ symbols indexed by the edges having endpoints in B^1 (in the sequel we will ignore the vertices in A that have less $(1-\alpha)\tau_1\Delta$ edges incident with vertices in B^1). Let $\delta_A(\mathsf{B})$ be:

1. a $\cup$-convex continuous function of B, and —
2. a lower bound on the minimum relative binary weight of a codeword of $\mathcal{C}_A$ under the restriction that the average binary weight of $\tau_1\Delta(1-\alpha)$ non-zero fixed symbols of it is at least $\mathsf{B}\ell$.

Lemma 4.4.5 *Let $\boldsymbol{c}$ be a codeword of $\mathbb{C}$ and let S, $|S| = \sigma n$, and Γ be the quantities defined above. The relative binary weight $\mathsf{w_b}(\boldsymbol{c})$ satisfies*

$$\mathsf{w_b}(\boldsymbol{c}) \geq \sigma\delta_A(\Gamma) + o(1) \ .$$

Proof. The proof of this lemma is similar to the proof of Lemma 8 in [9]. We rewrite the proof with the required adjustments.

From the definition of the function $\delta_A(\cdot)$,

$$\begin{aligned} \mathsf{w_b}(\boldsymbol{c}) &= \frac{1}{n}\sum_{v\in A}\mathsf{w_b}((\boldsymbol{c})_{E(v)}) \ = \ \frac{1}{n}\sum_{v\in S}\mathsf{w_b}((\boldsymbol{c})_{E(v)}) \\ &\geq \frac{\sigma}{|S|}\sum_{v\in S\setminus S_\alpha}\mathsf{w_b}((\boldsymbol{c})_{E(v)}) \ \geq \ \frac{\sigma}{|S|}\sum_{v\in S\setminus S_\alpha}\delta_A(\mathsf{B}_v) \ . \end{aligned}$$

Since $\mathsf{B}_v \geq \Gamma_v$ and $\delta_A(\cdot)$ is a non-decreasing function, we have $\delta_A(\mathsf{B}_v) \geq \delta_A(\Gamma_v)$. Therefore,

$$\begin{aligned} \frac{1}{|S|}\sum_{v\in S\setminus S_\alpha}\delta_A(\mathsf{B}_v) &\geq \frac{1}{|S|}\sum_{v\in S\setminus S_\alpha}\delta_A(\Gamma_v) \ = \ \frac{1}{|S|}\left(\sum_{v\in S}\delta_A(\Gamma_v) - \sum_{v\in S_\alpha}\delta_A(\Gamma_v)\right) \\ &= \frac{1}{|S|}\left(\sum_{v\in S}\delta_A(\Gamma_v)\right) - o_\Delta(1) \ \geq \ \delta_A\left(\sum_{v\in S}\frac{\Gamma_v}{|S|}\right) - o_\Delta(1) \\ &= \delta_A(\Gamma + o_\Delta(1)) - o_\Delta(1) = \delta_A(\Gamma) \pm o_\Delta(1) \ , \end{aligned}$$

where the second equality follows due to the fact that $1-S_\alpha/n = o_\Delta(1)$, the second inequality is due to convexity of $\delta_A(\cdot)$, the penultimate equality is due to Lemma 4.4.2, and the last equality is due to the uniform continuity of $\delta_A(\cdot)$. □

Lemma 4.4.6 *For any $\varepsilon > 0$, for Δ and ℓ large enough, for any r_A, there exist code $\mathcal{C}_A$ of rate r_A such that, for every B, $\delta_A(\mathsf{B})+\varepsilon$ is greater than any convex function that does not exceed $\omega_0(\mathsf{B})$, where $\omega_0(\mathsf{B})$ is the root of the equation in ω_0*

$$1 - r_A = \max_{r_A\omega_1+(1-r_A)\omega_2=\omega_0} r_A\omega_1\frac{\mathsf{H}_2(\mathsf{B})}{\mathsf{B}} + (1-r_A)\mathsf{H}(\omega_2) \ , \tag{4.27}$$

where ω_0 and ω_1 are constrained by $\mathsf{H}_2^{-1}(1-r_A) \leq \omega_0 \leq \mathsf{B}$ and $\omega_1 \leq \mathsf{B}$.

The claim of this lemma is not related to the definition of the code $\mathbb{C}$. This lemma appears as Lemma 13 in [9], and its proof appears therein as well.

In [9], the authors optimize the equality (4.27) over the values of ω_1 and ω_2. We will omit these technical calculations, but the reader can refer to [9, Section IV.B] for more details. The authors obtain in [9] that a lower bound $\omega^{\star\star}(\mathsf{B})$ on $\delta_A(\mathsf{B}) = \delta_A(\mathsf{B}, r_A)$ is given by

$$\omega^{\star\star}(\mathsf{B}) = r_A\mathsf{B} + (1 - r_A)\mathsf{H}_2^{-1}\left(1 - \frac{r_A}{1 - r_A}\mathsf{H}_2(\mathsf{B})\right) ,$$

for $\delta_{GV}(r_A) \le \mathsf{B} \le \mathsf{B}_1$, where B_1 is the only root of the equation

$$\delta_{GV}(r_A) = w^\star(\mathsf{B}) ,$$

and

$$w^\star(\mathsf{B}) = (1 - r_A)\left((2^{\mathsf{H}_2(\mathsf{B})/\mathsf{B}} + 1)^{-1} + \frac{\mathsf{B}}{\mathsf{H}_2(\mathsf{B})}\left(1 - \mathsf{H}_2\left((2^{\mathsf{H}_2(\mathsf{B})/\mathsf{B}} + 1)^{-1}\right)\right)\right) .$$

For $\mathsf{B}_1 \le \mathsf{B} \le \frac{1}{2}$, the function $\delta_A(\mathsf{B})$ is bounded from below by a tangent to the function $\omega^{\star\star}(\mathsf{B})$ drawn from the point $\left(\frac{1}{2}, \omega^\star(\frac{1}{2})\right)$.

The next theorem provides a lower bound on the relative minimum distance of the code $\mathbb{C}$. It follows by combining the results in Lemmas 4.4.3, 4.4.4, and 4.4.5, with the expression for $\delta_A(\mathsf{B}, r_A)$.

Theorem 4.4.7 *(This theorem is a counterpart of Theorem 14 in [9].) There exists a polynomial-time constructible family of binary linear codes $\mathbb{C}$ of length $N = n\Delta$, $n \to \infty$, and sufficiently large but constant $\Delta = \Delta(\varepsilon)$, whose relative minimum distance satisfies*

$$\delta(\mathcal{R}) \ge \max_{\mathcal{R} \le r_A \le 1} \left\{ \min_{\delta_{GV}(r_A) \le \mathsf{B} \le 1/2} \left(\delta_A(\mathsf{B}, r_A)\frac{1 - \mathcal{R}/r_A}{\mathsf{H}_2(\mathsf{B})}\right)\right\} - \varepsilon . \tag{4.28}$$

It follows from Theorem 4.4.7 that the generalized expander codes presented in this section are at least as good (from the point of view of their rate-distance trade-offs) as the codes in [9].

4.5 Discussion

Consider the code $\mathbb{C}$ as defined in Section 4.1, with parameter η slightly less than the right-hand side in inequality (4.3). We showed in Theorem 4.4.7, that the relative minimum distance $\delta(\mathcal{R})$ of such a code $\mathbb{C}$ of rate $\mathcal{R}$ is bounded from below by the expression in (4.28).

By contrast, consider the code $\mathbb{C}$ with parameter $\eta = 0$. For this code, the size of the set B^2 is zero, and therefore in such a case the code $\mathbb{C}$ coincides with the code $\mathbb{C}_{BZ2}$, defined in Section 1.4.3. The relative minimum distance δ_{BZ2} of that code of rate $\mathcal{R}$ is shown in [9] to satisfy

$$\delta_{BZ2}(\mathcal{R}) \geq \frac{1}{4}(1-\mathcal{R})^2 \cdot \min_{\delta_{GV}((1+\mathcal{R})/2)<\mathsf{B}<\frac{1}{2}} \frac{g(\mathsf{B})}{\mathsf{H}_2(\mathsf{B})} ,$$

as it was mentioned in Section 1.4.5.

In Table 4.1, which is taken from [9], we compare the values of these two bounds, $\delta(\mathcal{R})$ and $\delta_{BZ2}(\mathcal{R})$, for various values of $\mathcal{R}$, with the Zyablov bound $\delta_Z(\mathcal{R})$.

$\mathcal{R}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\delta_Z(\mathcal{R})$	0.129	0.073	0.044	0.026	0.015	0.008	0.0040	0.0015	0.00030
$\delta_{BZ2}(\mathcal{R})$	0.077	0.061	0.046	0.034	0.024	0.015	0.0084	0.0037	0.00089
$\delta(\mathcal{R})$	0.148	0.095	0.063	0.041	0.026	0.015	0.0078	0.0031	0.00073

Table 4.1: Two bounds on the relative minimum distance of expander codes in [9].

We can see that the bound $\delta(\mathcal{R})$ is superior to the Zyablov bound for any rate $\mathcal{R}$ in the table. It is also interesting to compare the bounds $\delta(\mathcal{R})$ and $\delta_{BZ2}(\mathcal{R})$. We see that the bound $\delta(\mathcal{R})$ is superior for low rates, while the bound $\delta_{BZ2}(\mathcal{R})$ is superior for high rates. It would be nice to derive a combined analytical bound which will be at least as good as both these bounds. One approach could be to take a value of η 'sliding' between zero and the right-hand side of (4.3), and to establish the point at which the value of η maximizes the appropriate value of the relative minimum distance of the corresponding code $\mathbb{C}$.

Unfortunately, we were not able to provide a good lower bound on the minimum distance of $\mathbb{C}$ for the intermediate values of η. The lower bound that we were able to derive was always less or equal to the highest bound among $\delta(\mathcal{R})$ and $\delta_{BZ2}(\mathcal{R})$. However, it seems that this research direction was not fully explored.

Chapter 5

Decoding Non-bipartite Expander Codes

5.1 Problem definition

In this chapter, we discuss the decoding of the codes of Sipser and Spielman [79], defined in Section 1.4.1 of this thesis. Recall that the relative minimum distance of the codes $\mathbb{C}$ (defined therein) is bounded from below by

$$\left(\frac{\delta_0 - \gamma_{\mathcal{G}}}{1 - \gamma_{\mathcal{G}}}\right)^2 = \delta_0^2 - O(\gamma_{\mathcal{G}}) \ , \tag{5.1}$$

where δ_0 is a relative minimum distance of the constituent code, and $\gamma_{\mathcal{G}}$ is a ratio between the second and the first largest eigenvalues of $A_{\mathcal{G}}$. On the other hand, the decoder in [79] is able to correct a number of errors that is close to only a fraction $\frac{1}{48}\delta_0^2$ of the code length.

In [84], Zémor improves on the fraction of correctable errors for the code $\mathbb{C}$. By taking an underlying graph $\mathcal{G}$ as a bipartite Ramanujan graph, Zémor is able to improve the fraction of correctable errors by a factor of 12, so this value became close to $\frac{1}{4}$ of the known lower bound on the minimum distance.

In Chapter 2, we further improve on the result of Zémor. By using erasures in addition to errors, and employing GMD-like decoding (see [30], [31]), we are able to improve the fraction of correctable errors by a factor of nearly 2, increasing it to (almost) half the known lower bound on the minimum distance of the code $\mathbb{C}$.

However, in Chapter 2 we use the same codes as Zémor does, namely the underlying graphs of these codes are bipartite. It is a question, however, whether this improvement can be achieved for low-complexity codes whose underlying graphs are not bipartite. In this section, we give a positive answer to this question. In particular, we show a reduction from the codes defined on non-bipartite graphs to codes defined on bipartite graphs. The latter codes can be decoded by the methods shown in Chapter 2 for codes based on bipartite graph, thus achieving almost half the minimum distance error correction. Thus, this reduction leads to the decoding of the original codes (defined on non-bipartite graphs) up to (almost) half (the lower bound on) the minimum distance.

5.2 Reduction

Consider the code $\mathbb{C} = (\mathcal{G}, \mathcal{C})$ as defined in Section 1.4.1, with a linear $[\Delta, k{=}r\Delta, d_0{=}\delta_0\Delta]$ constituent code $\mathcal{C}$ over $\mathbb{F} = \mathrm{GF}(q)$. Let $\mathcal{G} = (V, E)$ be an underlying graph and $A_{\mathcal{G}}$ its adjacency matrix. Let λ^* be the second largest absolute value of eigenvalue of $A_{\mathcal{G}}$, and denote $\gamma^*_{\mathcal{G}} = \lambda^*/\Delta$.

Below, we use the following graph construction. Define a new graph $\widehat{\mathcal{G}} = (\widehat{V}, \widehat{E})$, where for each vertex $v \in V$ we let $\widehat{V} = V_1 \cup V_2$ contain two *descendant* vertices v_1 and v_2, with $v_1 \in V_1$ and $v_2 \in V_2$ (thus, $|V_1| = |V_2| = |V| = n$). For each edge $e = a—b$ in E, we let $\widehat{E}$ contain the following two descendant edges:

$$e_1 = a_1—b_2 \ , \quad e_2 = a_2—b_1 \ .$$

Thus, every edge in $\widehat{E}$ has one endpoint in V_1 and one endpoint in V_2. The degree of every vertex in $\widehat{V}$ is Δ, and $|\widehat{E}| = 2|E| = n\Delta$.

It can be shown that if $\lambda_i \neq 0$ is an eigenvalue of $A_{\mathcal{G}}$ and its multiplicity is m_i, then λ_i and $-\lambda_i$ are both eigenvalues of the adjacency matrix of $\widehat{\mathcal{G}}$, $A_{\widehat{\mathcal{G}}}$, and their corresponding multiplicities are both m_i [56, p. 84]. It follows that $\gamma^*_{\mathcal{G}}$ is the ratio between the second largest eigenvalue of $A_{\widehat{\mathcal{G}}}$ and Δ.

An ordering on $\widehat{V}$ is assumed to be inherited from V, where v_i precedes v'_j (for $i, j \in \{1, 2\}$) if either $i < j$ or both $i = j$ and the parent vertex v precedes v' in V. This ordering induces an ordering on the set $\widehat{E}(v_i)$ of edges incident with v_i in $\widehat{\mathcal{G}}$.

Define the code $\widehat{\mathbb{C}}$ of length $n\Delta$ over $\mathbb{F}$, by means of the graph $\widehat{\mathcal{G}}$:

$$\widehat{\mathbb{C}} = \left\{ \boldsymbol{c} \in \mathbb{F}^{n\Delta} \ : \ (\boldsymbol{c})_{\widehat{E}(u)} \in \mathcal{C} \text{ for every } u \in \widehat{V} \ \right\} .$$

Essentially, the code $\widehat{\mathbb{C}}$ coincides with the code $\mathbb{C}$ defined in Section 2.1, for an appropriate choice of $\mathcal{C}$. Therefore, from Theorem 2.2.1, we obtain that the relative minimum distance $\widehat{\delta}$ of the code $\widehat{\mathbb{C}}$ satisfies

$$\widehat{\delta} \geq \delta_0 \cdot \frac{\delta_0 - \gamma^*_{\mathcal{G}}}{1 - \gamma^*_{\mathcal{G}}} \ . \tag{5.2}$$

Next, we define a mapping $\widehat{\varphi} : \mathbb{F}^{|E|} \to \mathbb{F}^{|\widehat{E}|}$, which copies each symbol of its argument twice, as follows. For $\boldsymbol{y} \in \mathbb{F}^{|E|}$,

$$(\widehat{\varphi}(\boldsymbol{y}))_{e_1} = (\widehat{\varphi}(\boldsymbol{y}))_{e_2} = y_e \; .$$

We also define a mapping $\widehat{\varphi}^{-1} : \mathbb{F}^{|\widehat{E}|} \to \mathbb{F}^{|E|}$, which keeps only half of symbols of its argument, as follows. For $\boldsymbol{z} \in \mathbb{F}^{\widehat{E}}$,

$$\left(\widehat{\varphi}^{-1}(\boldsymbol{z})\right)_e = z_{e_1} \; .$$

Now, we are ready to discuss a decoder for the code $\mathbb{C}$, which appears in Figure 5.1. The decoder applies the mapping $\widehat{\varphi}$ to its input $\boldsymbol{y}$, thus producing a word $\boldsymbol{z}$ of length Δn. Then, it applies a decoder $\mathcal{D}_{cont}$ for the code $\widehat{\mathbb{C}}$ to $\boldsymbol{z}$ (for example, $\mathcal{D}_{cont}$ can be a GMD decoder with any half min-distance decoder for the inner code $\mathcal{C}$, and the decoder $\mathcal{D}_\Phi$ in Figure 2.1 for the outer code, as described in Chapter 2.3). Such decoder $\mathcal{D}_{cont}$ is able to correct a fraction of errors up to

$$\delta_0 \cdot \frac{\delta_0/2 - \gamma^*_{\mathcal{G}}}{1 - \gamma^*_{\mathcal{G}}}$$

of the length of the code $\widehat{\mathbb{C}}$. Finally, the decoder applies the mapping $\widehat{\varphi}^{-1}$ to the result.

```
Input:  Received word $\boldsymbol{y} = (y_e)_{e \in E}$ in $\mathbb{F}^{|E|}$.
Let $\boldsymbol{z} \leftarrow \widehat{\varphi}(\boldsymbol{y})$.
Let $\boldsymbol{z} \leftarrow \mathcal{D}_{cont}(\boldsymbol{z})$.
Output:  $\widehat{\varphi}^{-1}(\boldsymbol{z})$ if there exists $\boldsymbol{c} \in \mathbb{C}$ such that $\boldsymbol{z} = \widehat{\varphi}(\boldsymbol{c})$ (and declare 'error' otherwise).
```

Figure 5.1: Decoder for the code $\mathbb{C}$ defined over a non-bipartite graph.

Theorem 5.2.1 *Let $\mathbb{C}$ be a code as considered in this section. Suppose that a word $\boldsymbol{y} \in \mathbb{F}^{|E|}$ is such that*

$$\mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) < \delta_0 \cdot \frac{\delta_0/2 - \gamma^*_{\mathcal{G}}}{1 - \gamma^*_{\mathcal{G}}} \cdot |E| \; . \tag{5.3}$$

for some codeword $\boldsymbol{c} \in \mathbb{C}$. Then, the decoder in Figure 5.1 when applied to the word $\boldsymbol{y}$, will output the word $\boldsymbol{c}$.

Proof. Consider a word $\boldsymbol{y}$ as given in the conditions of the theorem. Define the words

$$\boldsymbol{z} = \widehat{\varphi}(\boldsymbol{y}) \qquad \text{and} \qquad \widehat{\boldsymbol{c}} = \widehat{\varphi}(\boldsymbol{c}) \; .$$

Obviously,

$$\mathsf{d}(\boldsymbol{z}, \widehat{\boldsymbol{c}}) = 2 \cdot \mathsf{d}(\boldsymbol{y}, \boldsymbol{c}) < 2\delta_0 \cdot \frac{\delta_0/2 - \gamma^*_{\mathcal{G}}}{1 - \gamma^*_{\mathcal{G}}} \cdot |E| = \delta_0 \cdot \frac{\delta_0/2 - \gamma^*_{\mathcal{G}}}{1 - \gamma^*_{\mathcal{G}}} \cdot |\widehat{E}| \ , \tag{5.4}$$

where the first equality follows from the definition of $\widehat{\varphi}$, and the first inequality is due to the conditions of the theorem.

From the definition of $\widehat{\varphi}$, the word $\widehat{\boldsymbol{c}}$ is a codeword of $\widehat{\mathbb{C}}$. Note that the word $\boldsymbol{z}$ lies in the ball around the codeword $\widehat{\boldsymbol{c}}$ of radius less than the expression in the right-hand side of (5.4), which is the radius of correction for the decoder $\mathcal{D}_{cont}$. Therefore, application of this decoder on $\boldsymbol{z}$ will produce the word $\widehat{\boldsymbol{c}}$.

Finally, by the definition of $\widehat{\varphi}^{-1}$, $\boldsymbol{c} = \widehat{\varphi}^{-1}(\widehat{\boldsymbol{c}})$. Therefore, the decoder in Figure 5.1 will output the word $\boldsymbol{c}$. □

Conclusion. We presented a linear-time decoder for the codes $\mathbb{C}$ in [79] (defined in Section 1.4.1), which is able to correct a fraction of errors that is close to half the minimum distance of that code.

Remark. While the lower bound on the relative distance in (5.1) depends on $\gamma_{\mathcal{G}}$, both the bound on the relative minimum distance in (5.2) and the bound on the number of correctable errors in (5.3) depend on $\gamma^*_{\mathcal{G}}$, where $\gamma^*_{\mathcal{G}} \geq \gamma_{\mathcal{G}}$ (with a possible strict inequality for a non-bipartite $\mathcal{G}$).

Chapter 6

Asymptotic Goodness of Expander Codes with Weak Constituent Codes

6.1 Problem description

In this chapter, we consider the code $\mathbb{C} = (\mathcal{G}, \mathcal{C}_A : \mathcal{C}_B)$ defined as in Section 2.1. We discuss necessary and sufficient conditions on the relative minimum distances δ_A and δ_B of the constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$ so that the overall code $\mathbb{C}$ is asymptotically good, i.e. the relative minimum distance $\mathbb{C}$ is bounded away from zero as $n \to \infty$.

From Theorem 2.2.1, and from the properties of concatenated codes, the relative minimum distance of the code $\mathbb{C}$ is bounded from below by

$$\delta \geq \delta_A \cdot \frac{\delta_B - \gamma_{\mathcal{G}} \sqrt{\delta_B / \delta_A}}{1 - \gamma_{\mathcal{G}}} = \frac{\delta_A \delta_B - \gamma_{\mathcal{G}} \sqrt{\delta_A \delta_B}}{1 - \gamma_{\mathcal{G}}} \, .$$

It follows immediately that the relative minimum distance of the code $\mathbb{C}$ is bounded away from zero if

$$\delta_A \delta_B > \gamma_{\mathcal{G}} \sqrt{\delta_A \delta_B} \, ,$$

or, equivalently,

$$\sqrt{\delta_A \delta_B} > \gamma_{\mathcal{G}} \, . \tag{6.1}$$

We denote $d_A = \delta_A \Delta$ and $d_B = \delta_B \Delta$. Then, the Equation (6.1) can be rewritten as

$$\sqrt{d_A d_B} > \gamma_{\mathcal{G}} \Delta = \lambda \, , \tag{6.2}$$

where λ is the second largest eigenvalue of the adjacency matrix of $\mathcal{G}$.

However, as we show in the sequel, this condition is not always necessary. We aim at finding stricter conditions on δ_A and δ_B (or on d_A and d_B) for the code $\mathbb{C}$ to be asymptotically good. First, we survey some related results found in the literature.

Sipser and Spielman consider in [79] a code $\mathbb{C}$ with an underlying (α, ζ)-expander graph, where the code $\mathcal{C}_A$ is the repetition code and the code $\mathcal{C}_B$ is the parity code ($d_A = \Delta$, $d_B = 2$). They show that if $\zeta \geq \frac{3}{4}$, then the code $\mathbb{C}$ has a relative minimum distance at least α. Moreover, the authors present a decoding algorithm for that code, which is able to correct up to a number of errors which is a fraction $\alpha/2$ of the code length.

It is not known whether any Ramanujan graphs are suitable for the above Sipser-Spielman construction. While it is relatively straightforward to show that for Ramanujan graphs $\zeta \geq \frac{1}{4}$, Kahale in [43] shows that for Ramanujan graphs, ζ is bounded from below by approximately $\frac{1}{2}$. It is also shown in [43] that there exist Ramanujan graphs with expansion factor ζ equal to approximately $\frac{1}{2}$, and thus, the eigenvalue approach cannot provide better bounds on ζ. On the other hand, it is shown in [79] that a random bipartite graph has $\zeta \geq \frac{3}{4}$ with probability close to 1.

In [9], Barg and Zémor consider codes $\mathbb{C}$ with $d_A \geq 3$, $d_B \geq 3$. They show that for a family of random graphs, there are such codes $\mathbb{C}$ with relative minimum distance bounded away from zero. Therefore, it is enough to take rather weak constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$, but rather a good family of the underlying graphs $\mathcal{G}$ in order to obtain an asymptotically good family of the codes $\mathbb{C}$.

6.2 Girth of regular graphs

In this section, we discuss the lengths of cycles in Ramanujan graphs.

Definiton A *girth* of a graph is the length of the shortest cycle in it.

We begin with the following lemma [19, Chapter IV.1].

Lemma 6.2.1 *For $g \geq 3$ and $d \geq 3$ let*

$$n_0(g,d) = \begin{cases} 1 + \frac{d}{d-2}\left((d-1)^{(g-1)/2} - 1\right) & \textit{if } g \textit{ is odd },\\ \frac{2}{d-2}\left((d-1)^{g/2} - 1\right) & \textit{if } g \textit{ is even }. \end{cases}$$

Then any graph with minimal degree d and girth g has at least $n_0(g,d)$ vertices.

Corollary 6.2.2 *For $\Delta \geq 3$ and $2n \geq \Delta + 1$, the girth $g(\mathcal{G})$ of any bipartite Δ-regular graph $\mathcal{G}$ on $2n$ vertices is at most*

$$g(\mathcal{G}) \leq \left\lfloor 2\log_{\Delta-1}\left(n \cdot (\Delta - 2) + 1\right) \right\rfloor \approx 2\log_{\Delta-1}(n) + 2\ . \tag{6.3}$$

The corollary follows from Lemma 6.2.1 by re-arrangement of the terms in each of the two expressions in the definition of $n_0(g,d)$ in the lemma. Selection of the smallest between the two expressions leads to the required result.

In the sequel, we will need the following simple lemma. Its proof is somehwat based on the same idea as the proof presented in [19] for the counterpart of Lemma 6.2.1.

Lemma 6.2.3 *For a Δ-regular graph $\mathcal{G} = (V, E)$ with $2n$ vertices, with $\Delta \geq 3$ and $2n \geq \Delta + 1$, at least one of the two following sets of conditions holds:*

1. *There exist a vertex $v \in V$ and two distinct simple cycles $\mathcal{L}_1$, $\mathcal{L}_2$ in $\mathcal{G}$, such that:*

   (i) The vertex v belongs to both cycles $\mathcal{L}_1$ and $\mathcal{L}_2$;

   (ii) The length of $\mathcal{L}_1$ is bounded from above by the right-hand side expression of the inequality in (6.3), and the number of edges of $\mathcal{L}_2$ that do not belong to $\mathcal{L}_1$ is bounded from above by
   $$\lfloor \log_{\Delta-1}(2n) \rfloor + 1 \; .$$

2. *There exist a vertex $v \in V$, two simple cycles $\mathcal{L}_1$, $\mathcal{L}_2$, and a path $\mathcal{P}$ in $\mathcal{G}$ ($\mathcal{P}$ could be one vertex and no edges), such that:*

   (i) The vertex v belongs to $\mathcal{L}_1$ and $\mathcal{P}$, the cycles $\mathcal{L}_1$, $\mathcal{L}_2$ and the path $\mathcal{P}$ are all edge-disjoint, and the path $\mathcal{P}$ connects the vertex v with the cycle $\mathcal{L}_2$.

   (ii) The length of $\mathcal{L}_1$ is bounded from above by the right-hand side expression of the inequality in (6.3), and the number of edges in $\mathcal{L}_2$ and $\mathcal{P}$ together is bounded from above by
   $$\lfloor 2\log_{\Delta-1}(2n) \rfloor + 2 \; .$$

Proof. Let $\mathcal{G} = (V, E)$ be the given graph. First, by using Corollary 6.2.2, there exists a simple cycle in $\mathcal{G}$ of length g_1, which is less than or equal to the right-hand side expression of the inequality in (6.3). We denote this cycle by $\mathcal{L}_1$, and its edges and vertices by

$$e_1 \; : \; v_1 - v_2 \; , \quad e_2 \; : \; v_2 - v_3 \; , \quad \cdots \; , \quad e_{g_1} \; : v_{g_1} - v_1 \; .$$

We also denote by $\mathcal{H}$ the graph with the vertex set V and the edge set $E_{\mathcal{H}}$, where $E_{\mathcal{H}}$ is obtained from E by removing the edges along the cycle $\mathcal{L}_1$. Then, the degree in $\mathcal{H}$ of every vertex in $\mathcal{L}_1$ is $\Delta - 2$, and the degree in $\mathcal{H}$ of any other vertex is Δ.

We apply the Breadth First Search (BFS) algorithm [26, Chapter 1] to $\mathcal{H}$, starting at v_1. The BFS algorithm produces a tree with a root v_1 (see [26, Chapter 1] for details). We stop the BFS algorithm either if one of the vertices $v_2, v_2, \cdots, v_{g_1}$ appeared in the produced tree, or if some vertex $u \in V$ (possibly $u = v_1$) appeared at least twice in the tree. We denote by

g_2 the height of the tree, i.e. the number of the edges along the longest path from v_1 to any leaf in the tree.

Suppose that the BFS algorithm has stopped. This may happen due to one of the two following reasons:

1. A vertex $v_i \in \{v_1, v_2, \cdots, v_{g_1}\}$ has been encountered. Then, there is a simple path in $\mathcal{H}$ from v_1 to v_i of length g_2. Thus, the simple cycle $\mathcal{L}_2$ in $\mathcal{G}$ is obtained by connecting this path to the shortest among the two paths:
$$v_i \; - \; v_{i+1} \; - \; v_{i+2} \; - \; \cdots \; - \; v_1 \; ,$$
and
$$v_i \; - \; v_{i-1} \; - \; v_{i-2} \; - \; \cdots \; - \; v_1 \; .$$
The number of edges in $\mathcal{L}_2$ that do not belong to $\mathcal{L}_1$ is g_2. Obviously, the vertex v_1 belongs to the both $\mathcal{L}_1$ and $\mathcal{L}_2$, and $\mathcal{L}_1 \neq \mathcal{L}_2$.

2. The vertex $u \in V \backslash \{v_1, v_2, \cdots, v_{g_1}\}$ has been encountered twice. This means that there are two different paths in $\mathcal{H}$ of length g_2 (at most) between v_1 and u. Then, these two paths, when connected together, form a (not necessarily simple) cycle of length at most $2g_2$. Obviously, it is possible to extract (from this possibly not simple cycle), a simple cycle $\mathcal{L}_2$ and a path $\mathcal{P}$ connecting between v_1 and $\mathcal{L}_2$. The total number of edges in $\mathcal{L}_2$ and $\mathcal{P}$ is at most $2g_2$. The cycles $\mathcal{L}_1$ and $\mathcal{L}_2$ and the path $\mathcal{P}$ are disjoint, and (together with the vertex $v = v_1$) satisfy the requirement (i) of the case 2 of the lemma.

We complete the proof by bounding from above the value of g_2, as we describe next. The algorithm will not stop while all the vertices in the produced tree are different. Thus, there are $\Delta - 2$ vertices at distance 1 from v_1, $(\Delta-2)(\Delta-1)$ vertices at distance 2, and so on, and $(\Delta-2)(\Delta-1)^{g_2-2}$ vertices at distance $g_2 - 1$. The algorithm will not stop until
$$1 + (\Delta-2) + (\Delta-2)(\Delta-1) + \cdots + (\Delta-2)(\Delta-1)^{g_2-2} \leq 2n \; .$$
We obtain,
$$1 + (\Delta-2) \cdot \frac{(\Delta-1)^{g_2-1} - 1}{(\Delta-1)-1} \leq 2n \; ,$$
thus yielding
$$g_2 \leq \log_{\Delta-1}(2n) + 1 \; .$$
The latter inequality yields claim *(ii)* of (both cases) of the lemma. □

The following lemma provides a lower bound on the girth of certain families of Ramanujan graphs.

Lemma 6.2.4 *The girth of any of the bipartite Ramanujan graphs presented in [53], [62] is bounded from below by* $g(\mathcal{G}) \geq \frac{4}{3}\log_{\Delta-1}(2n)$.

The proof of this lemma can be found in [53] and [62].

6.3 Constituent codes with minimum distance 2

In this section, we consider 'the weakest case', — a code $\mathbb{C}$ with both $\mathcal{C}_A$ and $\mathcal{C}_B$ having minimum distance 2 and redundancy 1. Below, we distinguish between the case where $\mathcal{C}_A$ and $\mathcal{C}_B$ are parity codes over GF(2), and the case where $\mathcal{C}_A$ and $\mathcal{C}_B$ are codes of minimum distance 2 (and redundancy 1) over GF(q) ($q > 2$). We show that in both cases, the code $\mathbb{C}$ cannot be asymptotically good and we provide rather tight bounds on its minimum distance.

The next two theorems provide upper bounds on the minimum distance of $\mathbb{C}$ where the codes $\mathcal{C}_A$ and $\mathcal{C}_B$ are taken as the parity codes over GF(2), and as codes of minimum distance 2 (and redundancy 1) over GF(q), respectively.

Theorem 6.3.1 *Consider a code* $\mathbb{C}$ *defined as in Section 2.1, where* $\mathcal{C}_A$ *and* $\mathcal{C}_B$ *are taken as the parity codes of length* Δ *over* GF(2), *and* $\mathcal{G}$ *is a* Δ*-regular bipartite graph,* $\Delta \geq 3$. *Then, the minimum distance* D *of such a code* $\mathbb{C}$ *is bounded from above by*

$$D \leq \left\lfloor 2\log_{\Delta-1}\left(n\cdot(\Delta-2)+1\right)\right\rfloor .$$

Proof. Since the graph is bipartite and simple with n vertices on each side, it follows that $n \geq \Delta$, and so $2n \geq \Delta + 1$. Take the shortest cycle $\mathcal{L}$ in the graph $\mathcal{G}$ (the length g of $\mathcal{L}$ is even since $\mathcal{G}$ is bipartite). Define a word $\boldsymbol{c} \in \mathbb{F}^{|E|}$ as follows:

$$c_e = \begin{cases} 0 & \text{if } e \notin \mathcal{L} \\ 1 & \text{if } e \in \mathcal{L} \end{cases} .$$

Since g is even it follows that each vertex in $\mathcal{G}$ is incident with an even number of edges of $\mathcal{L}$. Therefore, $\boldsymbol{c} \in \mathbb{C}$.

The word $\boldsymbol{c}$ is such that the number of non-zero symbols in it is equal to the length of $\mathcal{L}$. By Corollary 6.2.2 we have that D is bounded from above by the right-hand side expression of the inequality in (6.3). □

Theorem 6.3.2 *Let a code* $\mathbb{C}$ *be defined as in Section 2.1, where* $\mathcal{C}_A$ *and* $\mathcal{C}_B$ *are taken as codes with minimum distance 2 and redundancy 1 of length* Δ *over* GF(q), *and* $\mathcal{G}$ *is a* Δ*-regular bipartite graph,* $\Delta \geq 3$. *Then, the minimum distance* D *of such a code* $\mathbb{C}$ *is bounded* *from above by*

$$D \leq \left\lfloor 2\log_{\Delta-1}\left(n \cdot (\Delta-2)+1\right) \right\rfloor + \left\lfloor 2\log_{\Delta-1}(2n) \right\rfloor + 2 \approx 4\log_{\Delta-1}(n) + 6\ .$$

Proof. It is given that both codes $\mathcal{C}_A$ and $\mathcal{C}_B$ have minimum distance 2 and redundancy 1. Then, there exist parity-check matrices for these codes consisting of one row each.

As before, the graph is bipartite with n vertices on each side, thus yielding that $n \geq \Delta$, and so $2n \geq \Delta + 1$. Therefore, the conditions of Lemma 6.2.3 are satisfied, and it follows that at least one of the two sets of conditions in the lemma are valid.

- Suppose that the first set (set 1) of the conditions in the lemma is valid. Then, let the cycles $\mathcal{L}_1$ and $\mathcal{L}_2$ be as guaranteed by the lemma.

  We define a word $\boldsymbol{c} \in \mathbb{C}$ as follows:

  $$c_e = 0 \text{ for } e \notin \mathcal{L}_1 \cup \mathcal{L}_2\ .$$

  In addition, we have to set the values of any c_e such that $e \in \mathcal{L}_1 \cup \mathcal{L}_2$. These values are constrained such that for every vertex $v \in \mathcal{L}_1 \cup \mathcal{L}_2$ either $(\boldsymbol{c})_{E(v)} \in \mathcal{C}_A$ (if $v \in A$) or $(\boldsymbol{c})_{E(v)} \in \mathcal{C}_B$ (if $v \in B$). We obtain a system of linear equations: the number of variables in it is equal to the number of edges in $\mathcal{L}_1 \cup \mathcal{L}_2$, and the number of equations is equal to the number of vertices in $\mathcal{L}_1 \cup \mathcal{L}_2$. Therefore, the number of variables is larger than the number of equations. There must exist a non-trivial solution for this system.

  The word $\boldsymbol{c}$ as above is such that the number of non-zero symbols in it is less than or equal to the number of edges in $\mathcal{L}_1 \cup \mathcal{L}_2$. By Lemma 6.2.3, this number is bounded from above by

  $$\left\lfloor 2\log_{\Delta-1}\left(n \cdot (\Delta-2)+1\right) \right\rfloor + \left\lfloor \log_{\Delta-1}(2n) \right\rfloor + 1\ . \tag{6.4}$$

- Now, suppose that the second set (set 2) of the conditions in the lemma is valid. The proof proceeds very similarly to the previous case. Namely, let the cycles $\mathcal{L}_1$ and $\mathcal{L}_2$, and the path $\mathcal{P}$ be as guaranteed by the lemma. We define a word $\boldsymbol{c} \in \mathbb{C}$ such that

  $$c_e = 0 \text{ for } e \notin \mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{P}\ ,$$

  and, in addition, we have to set the values of c_e for $e \in \mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{P}$. Similarly, we obtain a system of linear equations with the number of variables larger than the number of equations. There must exist a non-trivial solution for this system.

The obtained word $\boldsymbol{c}$ is such that the number of non-zero symbols in it is less than or equal to the number of edges in $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{P}$, and by Lemma 6.2.3, it is bounded from above by

$$\left\lfloor 2\log_{\Delta-1}\left(n\cdot(\Delta-2)+1\right)\right\rfloor + \left\lfloor 2\log_{\Delta-1}(2n)\right\rfloor + 2 \;. \tag{6.5}$$

Finally, we get that there exists a word $\boldsymbol{c} \in \mathbb{C}$ such that the number of non-zero symbols in it is bounded from above by either (6.4) or (6.5). By taking the highest among the two bounds, the required conclusion is obtained. □

6.4 Lower bounds on the minimum distance

The next theorem provides a lower bound on the minimum distance of the code $\mathbb{C}$ with both $\mathcal{C}_A$ and $\mathcal{C}_B$ having minimum distance 2, for certain types of underlying Ramanujan expanders.

Theorem 6.4.1 *Let a code $\mathbb{C}$ be defined as in Section 2.1, where $\mathcal{C}_A$ and $\mathcal{C}_B$ are taken as codes of length Δ over $\mathrm{GF}(q)$ with minimum distance 2, and $\mathcal{G} = (V, E)$ is a Δ-regular bipartite Ramanujan expander in [53], [62]. Then, the minimum distance of $\mathbb{C}$ is bounded from below by*

$$D \geq \frac{4}{3}\log_{\Delta-1}(2n) \;.$$

Proof. Let $\boldsymbol{c}$ be a non-zero word of $\mathbb{C}$, and let the edge set $Y \subseteq E$ be a support of $\boldsymbol{c}$, namely,

$$Y = \{e \in E \;:\; c_e \neq 0\} \;.$$

We define a subgraph $\mathcal{H}$ of $\mathcal{G}$ as follows: the edge set of $\mathcal{H}$ is Y, and the vertices of $\mathcal{H}$ are all endpoints of edges in Y. The degree of any vertex in the graph $\mathcal{H}$ is greater than or equal to 2. Consider one of the connected components of $\mathcal{H}$, denote it $\mathcal{H}_1$.

There exists a cycle in $\mathcal{H}_1$ — this can be shown by taking a vertex $v \in \mathcal{H}_1$ and making a walk in $\mathcal{H}_1$, starting at this v, and using each edge only once. The walk will stop at some vertex u either when the vertex u has been visited already, or when there are no unused edges available at u. Since the degrees of all vertices in $\mathcal{H}_1$ are ≥ 2, the walk will stop when entering the vertex u that has been visited already. We obtained the cycle in $\mathcal{H}_1$.

By using Lemma 6.2.4, the length of this cycle is at least $g(\mathcal{G}) \geq \frac{4}{3}\log_{\Delta-1}(2n)$. Therefore, the word $\boldsymbol{c}$ has at least $g(\mathcal{G})$ non-zero symbols (indexed by the edges of this cycle). We obtain that

$$\mathsf{w}(\boldsymbol{c}) \geq \frac{4}{3}\log_{\Delta-1}(2n) \;,$$

thus completing the proof. □

The following theorem presents a lower bound on the minimum distance of the code $\mathbb{C}$, when the constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$ have a small minimum distance. Obviously, the theorem provides a very weak bound, not strong enough to imply a sufficient condition for the asymptotic goodness of the code $\mathbb{C}$.

Theorem 6.4.2 *Consider the code $\mathbb{C}$ as above, with the constituent codes $\mathcal{C}_A$ and $\mathcal{C}_B$ of minimum distance $d_A \geq 2$ and $d_B \geq 2$, respectively, and the underlying graph is as presented in [53], [62]. Then, the minimum distance D of $\mathbb{C}$ is bounded from below by*

$$D \geq \Omega\left((2n)^{(1/3)\cdot \log_{\Delta-1}(d_A-1)(d_B-1)} - 1 \right) .$$

Proof. Pick a non-zero word $\boldsymbol{c} \in \mathbb{C}$, and let the edge set $Y \subseteq E$ be the support of $\boldsymbol{c}$. We define a subgraph $\mathcal{H}$ of $\mathcal{G}$ as follows: the edge set of $\mathcal{H}$ is Y, and the vertices of $\mathcal{H}$ are all endpoints of edges in Y. Since the shortest cycle in $\mathcal{G}$ has length at least $(4/3)\cdot\log_{\Delta-1}(2n)$, the same is true for the shortest cycle in the graph $\mathcal{H}$.

Pick some vertex v^0 in $\mathcal{H}$ which also belongs to A. We apply the Breadth First Search (BFS) algorithm [26, Chapter 1] to $\mathcal{H}$, starting at v^0. Since $(\boldsymbol{c})_{E(v^0)} \in \mathcal{C}_A$, the vertex v^0 has at least d_A neighbors in $\mathcal{H}$. We denote d_A of these neighbors as $v^1_1, v^1_2, \cdots, v^1_{d_A}$ (and ignore the rest of the neighbors of v^0). The BFS algorithm, when started at the vertex v^0, will search d_A edges entering these vertices.

Since $(\boldsymbol{c})_{E(v^1_i)} \in \mathcal{C}_B$, each v^1_i $(i \in 1, 2, \cdots, d_A)$ has at least $d_B - 1$ neighbors in $\mathcal{H}$ (besides v^0). Each such neighbor is a neighbor of exactly one vertex among the vertices v^1_i $(i \in 1, 2, \cdots, d_A)$ — otherwise there would be cycles of length 4 in $\mathcal{H}$. Therefore, there are $d_A \cdot (d_B - 1)$ vertices in $\mathcal{H}$ at distance 2 from the vertex v^0. The BFS algorithm, when started at vertex v^0, will search $d_A \cdot (d_B - 1)$ edges entering these vertices.

We can continue the same argument by induction on the distance d from v^0 as long as $d < 2/3 \cdot \log_{\Delta-1}(2n)$. For even d, there will be $d_A(d_A-1)^{d/2-1}(d_B-1)^{d/2}$ vertices at distance d from v^0 (and the same number of edges entering them); for odd d, there will be $d_A(d_A-1)^{(d-1)/2}(d_B-1)^{(d-1)/2}$ vertices at distance d from v^0 (and the same number of edges entering them). All the vertices that we encounter during this process, will be distinct from each other (otherwise, there would be a cycle of length $< 4/3 \cdot \log_{\Delta-1}(2n)$ in $\mathcal{H}$).

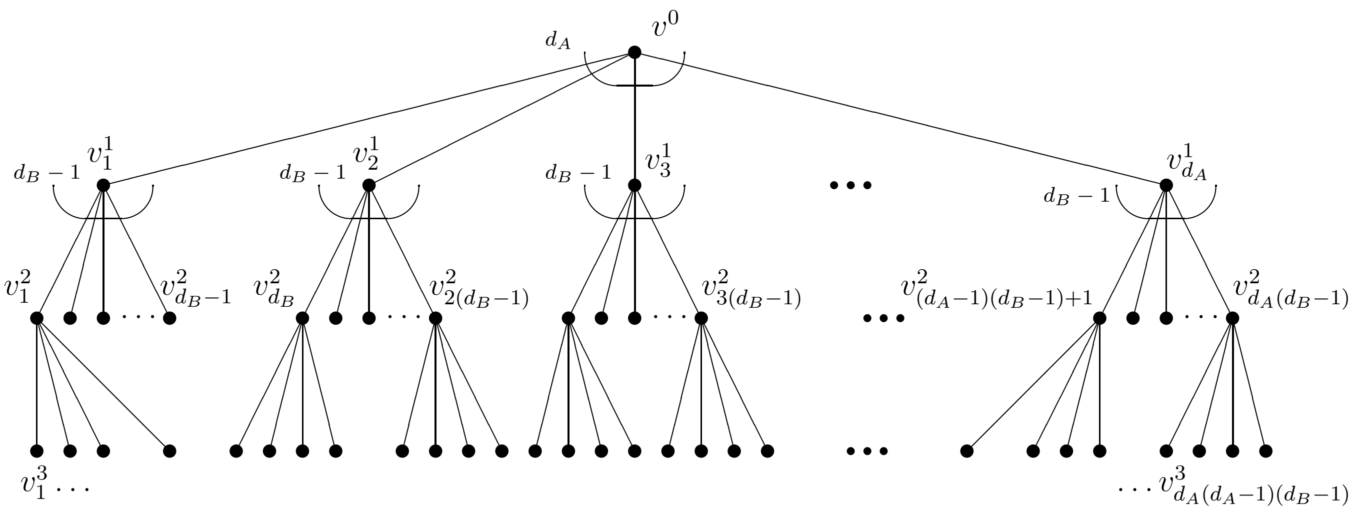

Figure 6.1: Illustration of proof of Theorem 6.4.2.

Assume that d is the largest even integer such that $d < (2/3) \cdot \log_{\Delta-1}(2n)$. The total number of edges in $\mathcal{H}$ that the BFS algorithm has passed through is bounded from below by

$$\begin{aligned} & d_A + d_A(d_B-1) + d_A(d_A-1)(d_B-1) + d_A(d_A-1)(d_B-1)^2 + \cdots \\ & \quad + \; d_A(d_A-1)^{(d-1)/2}(d_B-1)^{(d-1)/2} + d_A(d_A-1)^{d/2-1}(d_B-1)^{d/2} \\ & = \; d_Ad_B + d_Ad_B(d_A-1)(d_B-1) + \cdots + d_A(d_A-1)^{d/2-1}d_B(d_B-1)^{d/2-1} \\ & = \; d_Ad_B \cdot \frac{((d_A-1)(d_B-1))^{d/2} - 1}{(d_A-1)(d_B-1) - 1} \; . \end{aligned}$$

Recall that the edges in $\mathcal{H}$ correspond to non-zero symbols in $\boldsymbol{c}$, and thus we have obtained a lower bound on the number of such symbols. We substitute $d \approx (2/3) \cdot \log_{\Delta-1}(2n)$ to obtain that

$$\begin{aligned} D & \geq \; d_Ad_B \cdot \frac{((d_A-1)(d_B-1))^{1/3 \cdot \log_{\Delta-1}(2n)} - 1}{(d_A-1)(d_B-1) - 1} \\ & > \; ((d_A-1)(d_B-1))^{1/3 \cdot \log_{\Delta-1}(2n)} - 1 \\ & = \; \left(((d_A-1)(d_B-1))^{\log_{(d_A-1)(d_B-1)}(2n)} \right)^{1/3 \cdot \log_{\Delta-1}(d_A-1)(d_B-1)} - 1 \\ & = \; (2n)^{1/3 \cdot \log_{\Delta-1}(d_A-1)(d_B-1)} - 1 \; , \end{aligned}$$

as required. The multiplicative constant in the result follows from the approximation $d \approx \frac{2}{3} \log_{\Delta-1}(2n)$. □

It might be interesting to compare the lower bound in Theorem 6.4.2 with its counterpart in Theorem 2.2.1. We can see that Theorem 6.4.2 provides a lower bound even for 'weak' codes $\mathcal{C}_A$ and $\mathcal{C}_B$ that do not satisfy the condition (6.2), while Theorem 2.2.1 does not provide any non-trivial bound in that case.

Next, we consider the generalization of the code $\mathbb{C}$ as follows: for any vertex $v \in B$ we allow $\deg_{\mathcal{G}}(v)$ be different from Δ. Thus, for vertices $v \in B$ we use (possibly different) constituent codes $\mathcal{C}_B(v)$ of length $|E(v)|$, and so,

$$\mathbb{C} = \Big\{ \boldsymbol{c} \in \mathbb{F}^{|E|} \ : \ (\boldsymbol{c})_{E(v)} \in \mathcal{C}_A \text{ for every } v \in A \\ \text{and } (\boldsymbol{c})_{E(v)} \in \mathcal{C}_B(v) \text{ for every } v \in B \Big\} \ . \tag{6.6}$$

The theorem below presents a lower bound on the minimum distance of such code $\mathbb{C}$, that take into account the expansion coefficient ζ.

Theorem 6.4.3 *Let $\mathcal{G} = (V, E)$ be a bipartite (α, ζ)-expander graph with a vertex set $V = A \cup B$, $A \cap B = \emptyset$, such that every edge in E had one endpoint in A and one in B. For every vertex $u \in A$ let $\deg_{\mathcal{G}}(u) = \Delta$, and let $\delta_A > 1 - \zeta$. Suppose that $\mathcal{C}_A$ is a linear code of length Δ and relative minimum distance δ_A over $\mathbb{F}$, and codes $\mathcal{C}_B(v)$ (for every vertex $v \in B$) are codes of lengths $|E(v)|$ and minimum distance greater than or equal d_B over $\mathbb{F}$. Let $\mathbb{C}$ be a code defined in (6.6) with respect to the graph $\mathcal{G}$ and the codes $\mathcal{C}_A$ and $\mathcal{C}_B(v)$ as above. If*

$$\frac{\delta_A}{\zeta + \delta_A - 1} < d_B \ , \tag{6.7}$$

then the relative minimum distance of $\mathbb{C}$ is bounded from below by $\alpha\delta_A$.

Proof. Consider a non-zero codeword $\boldsymbol{c} \in \mathbb{C}$, and let Y be the support of $\boldsymbol{c}$. Let $S \subseteq A$ be the set of vertices that are endpoints of edges in Y, and let $\sigma = |S|/n$. We assume (by contradiction) that $\sigma \leq \alpha$, and (in the sequel) derive a necessary condition on the parameters of the code $\mathbb{C}$. This condition will yield the required bound.

Let $T \subseteq B$ be the set of neighbors of the vertices in S. Denote the sets T_0 and T_1 as

$$T_0 = \{ v \in T \ : \ c_e = 0 \text{ for all } e \in E(v) \} \ ,$$

and

$$T_1 = T \backslash T_0 \ .$$

Let K be the fraction of non-zero edges (edges that are the indexes of the non-zero symbols in $\boldsymbol{c}$) in the graph $\mathcal{G}_{S \cup T}$ (induced from $\mathcal{G}$ by the vertex set $S \cup T$), namely

$$\mathsf{K} = \frac{|Y|}{|E_{S \cup T}|} \ .$$

Then, obviously, $\mathsf{K} \geq \delta_A$.

The total number of edges in the graph $\mathcal{G}_{S \cup T}$ is $\sigma \Delta n$. The number of edges e in $\mathcal{G}_{S \cup T}$ such that $c_e = 0$ is

$$\sigma \Delta n \cdot (1 - \mathsf{K}) \ .$$

Every vertex in T_0 has at least one zero edge in $E_{S \cup T}$ incident with it, and therefore the number of zero edges is at least $|T_0|$. Thus,

$$\sigma \Delta n \cdot (1 - \mathsf{K}) \geq |T_0| \ . \tag{6.8}$$

From the graph expansion, and using the assumption $\sigma \leq \alpha$, we have that

$$\sigma \Delta n \cdot \zeta \leq |T| \ . \tag{6.9}$$

By combining (6.8) and (6.9), we obtain

$$|T_1| = |T| - |T_0| \geq \sigma \Delta n \zeta - \sigma \Delta n (1 - \mathsf{K}) = \sigma \Delta n (\zeta + \mathsf{K} - 1) > 0 \ .$$

The number of non-zero edges in $E_{S \cup T}$ is $\sigma \Delta n \mathsf{K}$. Therefore, an 'average' vertex in T_1 has $\sigma \mathsf{K} \Delta n / |T_1|$ non-zero edges incident with it. However,

$$\frac{\sigma \mathsf{K} \Delta n}{|T_1|} \leq \frac{\sigma \mathsf{K} \Delta n}{\sigma (\zeta + \mathsf{K} - 1) \Delta n} = \frac{\mathsf{K}}{\zeta + \mathsf{K} - 1} \ . \tag{6.10}$$

Therefore, there exists a vertex in T_1 that has at most $\frac{\mathsf{K}}{\zeta + \mathsf{K} - 1}$ non-zero edges incident with it. This is impossible if

$$\frac{\mathsf{K}}{\zeta + \mathsf{K} - 1} < d_B \ . \tag{6.11}$$

Next, note that the function

$$f(\mathsf{K}) = \frac{\mathsf{K}}{\zeta + \mathsf{K} - 1} = 1 + \frac{1 - \zeta}{\zeta + \mathsf{K} - 1}$$

is monotonically decreasing in K. Therefore, from (6.7),

$$d_B > \frac{\delta_A}{\zeta + \delta_A - 1} \geq \frac{\mathsf{K}}{\zeta + \mathsf{K} - 1} \ ,$$

in contradiction to the assumption that $\sigma \leq \alpha$.

We obtain that if condition (6.7) holds, then $\sigma > \alpha$. Moreover, every vertex in S has at least a fraction δ_A of edges corresponding to the non-zero symbols in $\boldsymbol{c}$. Therefore, the relative minimum distance of $\mathbb{C}$ is bounded from below by $\alpha \delta_A$. □

Example 6.4.1 Consider the case where $d_B = 2$, $\delta_A = 1$, and $\zeta > \frac{1}{2}$. In this case, condition (6.7) is met, and therefore the corresponding code $\mathbb{C}$ is good asymptotically. □

Example 6.4.1 appears as a theorem for LDPC codes in [72, Chapter 8].

Example 6.4.2 Take a Ramanujan graph as in [53], [62], in which case ζ is (very close to) $\frac{1}{2}$. Pick $d_B = 3$ and $\delta_A = 1$. In this case, again, condition (6.7) is met, and, therefore, the relative minimum distance of $\mathbb{C}$ is at least $\alpha\delta_A$.

We compare condition (6.7) with condition (6.2). For the present example, Theorem 6.4.3 yields asymptotic goodness of the code $\mathbb{C}$, while Theorem 2.2.1 does not. □

Example 6.4.3 Take a Ramanujan graph as in [53], [62], in which case ζ is (almost) $\frac{1}{2}$. In this example we would like to take a code $\mathcal{C}_A$ with a relatively high minimum distance. For non-trivial *binary* codes $\mathcal{C}_A$ it holds that $\delta_A < \frac{1}{2}$. However, we can take $\mathcal{C}_A$ over $\mathbb{F} = \mathrm{GF}(2^2)$. There are such codes $\mathcal{C}_A$ having δ_A close to $\frac{3}{4}$. In addition, take $\mathcal{C}_B$ to have $d_B = 5$.

For this selection of parameters condition (6.7) is satisfied, and the corresponding code $\mathbb{C}$ is good asymptotically over $\mathbb{F}$. Next, we can consider the code $\mathbb{C}$ over GF(2) rather than over $\mathbb{F}$. Thus, each symbol over $\mathbb{F}$ becomes a pair of binary bits. It is easy to see that if the code $\mathbb{C}$ is asymptotically good over $\mathbb{F}$, it will also be asymptotically good over GF(2). Such a code $\mathbb{C}$ can be viewed as a binary code defined over the graph obtained from a Ramanujan graph by duplicating each edge twice.

It is worth mentioning that for a selection of the code $\mathbb{C}$ as above, Theorem 2.2.1 does not provide any non-trivial lower bound. The approach described in the present example can be generalized toward large fields, thus producing asymptotically good code families for any extension field of GF(2). In contrast, Theorem 2.2.1 does not provide any significant bound on their minimum distance. From this observation, we believe that (in some cases) stronger bounds on the minimum distance could be obtained when using the bounds on the girth and the expansion properties of Ramanujan graphs, in addition to using separation between the eigenvalues of the adjacency matrix. □

As we see, in some cases Theorem 6.4.3 improves over Theorem 2.2.1 even for Ramanujan graphs. However, the advantage of Theorem 6.4.3 becomes more evident for expanders that have high values of ζ. There are known constructions for such graphs, in particular using a so-called *zig-zag construction* (see [68]). The best such construction known to date appears in [21]. For those expanders, the expansion factor ζ can be as close to 1 as desired (by paying the price through large values of Δ). As for $\gamma_{\mathcal{G}}$, for zig-zag constructions its value is believed to be larger than for Ramanujan graphs. Thus, in [68], $\gamma_{\mathcal{G}}$ is shown to achieve $O(\frac{1}{\Delta^{1/3}})$ (at most).

In Appendix F, we present additional sufficient conditions on the asymptotic goodness of the code $\mathbb{C}$ where the graph $\mathcal{G}$ is Δ-regular (although some of those bounds can be shown also for the case where the degrees of the vertices in the set B are not all equal Δ). The proofs therein are somewhat different from the proof of Theorem 6.4.3. However, as we also show in Appendix F, these conditions are stronger compared with the condition in Theorem 6.4.3.

Chapter 7

Conclusions and Further Research

7.1 Summary

Over the last years, the practical success of LDPC codes has stimulated a lot of related research. The problem of explicit construction of provably good LDPC codes was solved by Sipser and Spielman by the invention of expander codes in [79], [81]. These codes were the first known codes to admit linear-time encoding and decoding that corrects a constant fraction of errors. Further works on expander codes, such as [8], [10], [11], [28], [29], [38], [75], [80], [84], have demonstrated the potential of expander-based code constructions. The expander codes were shown to be a key ingredient in nearly-MDS codes with the smallest known alphabet size (admiting a linear-time encoding and decoding) [38], [75].

However, the field of expander codes seems to be not fully explored yet. In the sequel, we will discuss the possible future directions for research on expander codes.

7.2 Our results

In this thesis, we presented several new constructions and bounds for expander codes. In this section, we summarize the results presented in the current thesis.

In Chapter 2, we improved on the known bounds on the parameters of expander codes, which were presented in a series of works of Barg and Zémor [8], [10], [84], and in a work of Guruswami and Indyk [38]. Thus, in a work, which preceded [8] (see [80]), we improved on the number of correctable errors for the codes in [84] by a factor of (approximately) 2.

After publication of the results in [8], we (slightly) improved on the lower bound on the minimum distance of expander codes therein. We showed that the codes therein can be viewed as a concatenation of a nearly-MDS expander code with an appropriate inner code. This nearly-MDS code admits a linear-time encoding and decoding, and has a smaller alphabet size compared to its counterpart presented in [38].

By employing this approach, we were able to present a new decoding algorithm for expander codes together with a novel analysis. We showed that our algorithm can correct (slightly) more errors than its counterpart in [8]. Moreover, the decoding time of our algorithm has only a polynomial dependence on the degree Δ of the underlying graph. In contrast, for the decoder in [8], this dependence may be exponential.

In Chapter 3, we investigated the decoding error probability of codes as a function of their block length. We showed that the existence of codes with polynomially small decoding error probability implies the existence of codes with exponentially small decoding error probability. Specifically, we assumed that there exists a family of codes of length N and rate $\mathcal{R} = (1-\varepsilon)\mathsf{C}$ (C is the capacity of a binary symmetric channel), whose decoding probability decreases inverse polynomially in N. Then, we showed that if the decoding probability decreases sufficiently fast, but still only inverse polynomially fast in N, then there exists another such family of codes whose decoding error probability decreases exponentially fast in N. Moreover, if the decoding time complexity of the assumed family of codes is polynomial in N and $1/\varepsilon$, then the decoding time complexity of the presented family is linear in N and polynomial in $1/\varepsilon$. We compared these codes to the codes of Barg and Zémor [8], [10]. We showed that the latter families cannot be tuned to have exponentially decaying (in N) error probability, and at the same time to have decoding time complexity linear in N and polynomial in $1/\varepsilon$.

In Chapter 4, we presented a family of so-called *generalized* expander codes. We showed that generalized expander codes have distance-rate trade-offs which are (asymptotically) at least as good as those of the codes in [8]. We presented a linear-time decoding algorithm for the generalized codes. Finally, using techniques as in [9], we showed that binary generalized expander codes have distance-rate trade-offs which (asymptotically) attain the parameters in [9].

In Chapter 5, we considered expander codes defined over non-bipartite graphs. The not-necessarily bipartite expander code model was first studied by Sipser and Spielman in [79]. We presented a reduction, which allows to decode those codes in linear time while correcting a number of errors up to (almost) half of the minimum distance of those codes. This improves a fraction of correctable errors in [79] by a factor of (approximately) 24.

In Chapter 6, we investigated expander codes with 'weak' constituent codes. We tried to answer the question: what are the weakest constituent codes such that the overall expander code family is asymptotically good? We found lower and upper bounds on the minimum distance of the expander codes having codes of minimum distance 2 as their constituent codes. In this case, we showed that the overall code cannot be asymptotically good. Finally, we derived some sufficient conditions on the parameters of the constituent codes, such that the overall expander code family is asymptotically good.

7.3 Future directions

In this section, we mention some of the interesting research problems related to expander and LDPC codes.

Distance–rate trade-offs. A trade-off between the rate and the relative minimum distance is one of the main characteristics of a code family. An explicitly constructible binary expander codes, which are decodable in linear time, were presented in [9]. The distance–rate trade-offs for these codes lie above the Zyablov bound. In Chapter 4, we achieve a similar result for slightly different family of expander-based codes. In that chapter, we also discussed some possible direction for a potential improvement of the bound in [9]. Further improvement of that bound would be a nice result in coding theory.

Alphabet size of nearly-MDS codes. In this thesis, we presented a construction of both linear-time encodable and decodable nearly-MDS expander codes of rate r and relative minimum distance δ with the size of the code alphabet given by

$$\exp\left\{\frac{1}{\epsilon^3}\log\frac{1}{\epsilon}\right\} ,$$

such that $r + \delta \geq 1 - \epsilon$ for any small ϵ. Generally, smaller alphabet sizes could result in faster encoding and decoding algorithms. It might be interesting to further reduce the size of the code alphabet. On the other hand, it would also be interesting to derive a lower bound on the alphabet size of codes that are based on expander graphs, or possibly Ramanujan-type expander graphs.

Other types of expander graphs. In this thesis, we presented several improvements on the bounds on the minimum distance and on the number of correctable errors of expander codes based on Ramanujan graphs. The techniques involved in the analysis were based on the eigenvalues properties of expander graphs. However, explicit constructions for other types of expanders were discovered recently, for example the the zig-zag construction in [21]. The construction therein has better vertex-expansion properties than Ramanujan graphs, but, on the other hand, their eigenvalue separation properties are not as good as those of the Ramanujan counterparts. It would be interesting to derive similar bounds on the minimum distance and on the decoding radius of linear-time decoders for expander codes based on non-Ramanujan expanders, in particular on the expanders in [21].

Generalized expander codes. In Chapter 4, we presented generalized expander codes. We showed that generalized expander code parameters are at least as good as the parameters of the expander codes in [9]. We also presented a decoding algorithm for those generalized expander codes. It might be interesting to further explore the properties of generalized expander codes. An interesting question to answer is whether generalized expander codes have any advantage over the known expander codes, similarly to the strength of irregular LDPC codes compared with regular LDPC codes.

Expander codes with weak congtituent codes. In Chapter 6, we presented several necessary conditions for the asymptotic goodness of expander codes. However, the main question formulated in that chapter was not answered in full. That is, what are the weakest constituent codes such that the overall expander code is asymptotically good? To answer that question, further investigation should be done.

Minimum pseudo-code weight. It is known that the *minimum pseudo-code weight* [47] is closely related to the minimum code distance. Building codes with a good minimum pseudo-code weight leads to good LDPC codes. Recently, some work on the minimum pseudo-code weight of expander codes was done by Kelley and Sridhara in [44]. In particular, the authors derived some bounds on the minimum pseudo-code weight of expander codes over the BEC and BSC channels. No such bound has been derived yet over the AWGN channel. These bounds over the BEC and BSC can be slightly improved, using techniques presented in this thesis. It would also be interesting to obtain similar bounds over the AWGN channel, and probably to obtain more extensive characteristics of pseudo-weight distribution of expander codes, since this can probably improve our understanding of LDPC codes.

Study of small codes. The lower bounds on the number of correctable errors, as found in the literature, do apply to the asymptotic behavior of codes but do not adequately explain the behavior of codes of a given length. For such codes of small length, the corresponding bipartite graphs necessarily have small cycles, which the analysis of [55] cannot handle. The study of small codes can be useful for practical purposes. Some preliminary work on short LDPC codes constructed from expander graphs, was done in [42], but there is still much yet to be studied.

Constrained LDPC codes. Below we mention two important classes of constraints.

The (d, k)-runlength-limited (RLL) constraint, where between any two consecutive ones in a binary sequence there must appear at least d and at most k zeros. The properties of (d, k)-RLL sequences together with encoding-decoding methods were extensively studied in the literature: see for example [39, Chapters 4, 5], [60], [61].

The *dc*-free constraint, where the number of ones in every binary codeword is equal to the number of zeros. Codes that satisfy *dc*-free constraint were studied, for example, in [39, Chapters 9, 10].

Codes that satisfy these or other constraints belong to the class of constrained codes.

The subject of constrained low-density parity-check codes has not been extensively studied despite its great practical importance. The construction of efficient *dc*-free LDPC codes or run-length-limited constrained LDPC codes, that have good encoding and decoding algorithms, is another interesting research problem, that is to be explored. Using well-structured expander graphs could possibly simplify this problem.

Conclusions

In this thesis, we combined classical techniques from coding theory, like GMD-decoding, concatenated code analysis, and others, with expander-based constructions. We were able to obtain new constructions of linear-time encodable and decodable LDPC codes that have good trade-offs between their relative minimum distance and the code rate.

Bibliography

[1] N. Alon, *Eigenvalues and expanders, Combinatorica, 6 (1986), pp. 83–96.*

[2] N. Alon, J. Bruck, J. Naor, M. Naor, R. M. Roth, *Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs, IEEE Trans. Inform. Theory, 38 (1992), pp. 509–516.*

[3] N. Alon, J.H. Spencer, *The Probabilistic Method,* 2nd ed., Wiley, New York, 2000.

[4] N. Alon, J. Edmonds, M. Luby, *Linear time erasure codes with nearly optimal recovery, Proc. 36th Annual IEEE Symp. on Foundations of Computer Science (FOCS), Oct. 1995, pp. 512–519,* Milwaukee, Wisconsin.

[5] N. Alon, M. Luby, *Linear time erasure-resilient code with nearly optimal recovery, IEEE Trans. Inform. Theory, 42 (1996), pp. 1732–1736.*

[6] A. Ashikhmin, V. Skachek, *Decoding of expander codes at rates close to capacity, Proc. IEEE Int. Symposium on Inform. Theory (ISIT), 2005, pp. 317–321,* Adelaide, Australia.

[7] A. Ashikhmin, V. Skachek, *Decoding of expander codes at rates close to capacity, IEEE Trans. Inform. Theory, 52 (2006), pp. 5475–5485.*

[8] A. Barg, G. Zémor, *Concatenated codes: serial and parallel, IEEE Trans. Inform. Theory, 51 (2005), pp. 1625–1634.*

[9] A. Barg, G. Zémor, *Distance properties of expander codes, IEEE Trans. Inform. Theory, 52 (2006), pp. 78-90.*

[10] A. Barg, G. Zémor, *Error exponents of expander codes, IEEE Trans. Inform. Theory, 48 (2002), pp. 1725–1729.*

[11] A. Barg, G. Zémor, *Error exponents of expander codes under linear-complexity decoding, SIAM J. Discrete Math., 17 (2004), pp. 426–445.*

[12] C. Berrou, A. Glavieux, P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding, Proc. Int. Conf. Communication, May 1993, 1064–1070*, Geneva, Switzerland.

[13] Y. Bilu, S. Hoory, *On codes from hypergraphs, European J. of Combinatorics, 25(3) (2004), pp. 339–354.*

[14] M. Blaum, J. Bruck, A. Vardy, *MDS array codes with independent parity symbols, IEEE Trans. Inform. Theory, 42 (1996), pp. 529–542.*

[15] M. Blaum, J. Fan, L. Xu, *Soft decoding of several classes of array codes, Proc. IEEE Int. Symposium on Inform. Theory (ISIT), 2002,* Lausanne, Switzerland.

[16] M. Blaum, R.M. Roth, *New array codes for multiple phased burst correction, IEEE Trans. Inform. Theory, 39 (1993), pp. 66-77.*

[17] M. Blaum, R.M. Roth, *On lowest-density MDS codes, IEEE Trans. Inform. Theory, 45 (1999), pp. 46–59.*

[18] B. Bollobás, *Extremal Graph Theory,* Academic Press, London, 1978.

[19] B. Bollobás, *Modern Graph Theory,* Springer-Verlag, New York, 1998.

[20] D. Burshtein, G. Miller, *Expander graph arguments for message-passing algorithms, IEEE Trans. Inform. Theory, 47 (2001), pp. 782–790.*

[21] M. Capalbo, O. Reingold, S. Vadhan, A. Wigderson, *Randomness conductors and constant-degree lossless expanders, Proc. 34th Annual ACM Symposium on Theory of Computing (STOC), May 2002, pp. 659–668*, Montréal, Quebec, Canada.

[22] G. Davidoff, P. Sarnak, A. Valette, *Elementary Number Theory, Group Theory, and Ramanujan Graphs,* Cambridge University Press, Cambridge, UK, 2003.

[23] I. Dumer, *Concatenated codes and their multilevel generalizations,* in *Handbook of Coding Theory, Volume II, pp. 1911–1988,* V.S. Pless and W.C. Huffman (Editors), North-Holland, Amsterdam, Netherlands, 1998.

[24] P. Erdős, H. Sachs, *Reguläre graphen gegebener taillenweite mit minimaler knotenzahl, Wiss. Z. Univ. Halle Martin Luther Univ. Halle-Wittenberg Math.-Natur. Reine, 12 (1963), pp. 251–257.*

[25] T. Etzion, A. Trachtenberg, A. Vardy, *Which codes have cycle-free Tanner graphs?, IEEE Trans. Inform. Theory, 45 (1999), pp. 2173–2181.*

[26] S. Even, *Graph Algorithms,* Computer Science Press, Rockville, Maryland, 1979.

[27] J. Fan, *Array codes as low-density parity-check codes, 2nd Int. Symposium on Turbo Codes and Related Topics, Sept. 2000,* Brest, France.

[28] J. Feldman, T. Malkin, C. Stein, R.A. Servedio, M.J. Wainwright, *LP decoding corrects a constant fraction of errors,* to appear in *IEEE Trans. Inform. Theory, 52 (2006).*

[29] J. Feldman, C. Stein, *LP decoding achieves capacity, Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA), Jan. 2005, pp. 460–469,* Vancouver, BC, Canada.

[30] G.D. Forney, Jr., *Concatenated Codes,* M.I.T. Press, Cambridge, Massachusetts, 1966.

[31] G.D. Forney, Jr., *Generalized minimum distance decoding, IEEE Trans. Inform. Theory, 12 (1966), pp. 125–131.*

[32] H. Fujita, K. Sakaniwa, *Justesen-type modified expander codes and their decoding algorithm, IEICE Trans. Fund. Electronics, Comm. and Comp. Sciences E88-A(10) (2005), pp. 2708–2714.*

[33] R.G. Gallager, *Low density parity check codes, IRE Trans. Inform. Theory,* 8 (1962), pp. 21–28.

[34] R.G. Gallager, *Low-Density Parity-Check Codes,* M.I.T. Press, Cambridge, Massachusetts, 1963.

[35] R.G. Gallager, *Information Theory and Reliable Communication,* John Wiley & Sons, New York, 1968.

[36] V. Guruswami, P. Indyk, *Expander-based constructions of efficiently decodable codes, Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2001, pp. 658–667,* Las Vegas, Nevada.

[37] V. Guruswami, P. Indyk, *Linear-time codes to correct a maximum possible fraction of errors, Proc. 39th Annual Allerton Conference on Communication, Control, and Computing, 2001,* Monticello, Illinois.

[38] V. Guruswami, P. Indyk, *Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets, Proc. 34th Annual ACM Symposium on Theory of Computing (STOC), May 2002, pp. 812–821*, Montréal, Quebec, Canada.

[39] K.A.S. Immink, *Codes for Mass Data Storage Systems,* Shannon Foundation Publishers, The Netherlands, 1999.

[40] H. Janwa, K. Lal, *On Tanner codes: minimum distance and decoding, Appl. Algebra Eng. Comm. Comput., 13 (2003), pp. 335–347.*

[41] J. Justesen, *A class of constructive asymptotically good algebraic codes, IEEE Trans. Inform. Theory, 18 (1972), pp. 652–656.*

[42] J. Justesen, T. Hoeholdt, *From concatenated codes to graph codes, Proc. IEEE Inform. Theory Workshop (ITW), Oct. 2004,* San Antonio, Texas.

[43] N. Kahale, *Eigenvalues and expansion of regular graph, Journal of the ACM, 42 (1995), pp. 1091–1106.*

[44] C. Kelley, D. Sridhara, *Eigenvalues bounds on the minimum pseudocodeword weight of expander codes,* submitted to *Appl. Algebra Eng. Comm. Comput., Aug. 2006.*

[45] A. Khandekar, R.J. McEliece, *On the complexity of reliable communication on the erasure channel, Proc. IEEE International Symposium on Information Theory (ISIT), June 2001, p.1,* Washington, DC.

[46] S. Kim, *Generalized minimum distance iterative decoding of Tanner codes, IEEE Comm. Letters, 8 (2005), pp. 738–740.*

[47] R. Koetter, P. Vontobel, *Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes,* submitted to *IEEE Trans. Inform. Theory, Dec. 2005*, also available at `http://www.arxiv.org/abs/cs.IT/0512078`.

[48] Y. Kou, S. Lin, M.P.C. Fossorier, *Low-density parity-check codes based on finite geometries: a rediscovery and new results, IEEE Trans. Inform. Theory, 47 (2001), pp. 2711–2736.*

[49] F. R. Kschischang, B. J. Frey, *Iterative decoding of compound codes by probability propagation in graphic models, IEEE J. Select. Areas Commun., 16 (1998), pp. 219–230.*

[50] R. Lidl, H. Niederreiter, *Finite Fields.* Cambridge Univ. Press, Cambridge, UK, 1984.

[51] S. Litsyn, V. Shevelev, *Distance distributions in ensembles of irregular low-density parity-check codes, IEEE Trans. Inform. Theory, 49 (2003), pp. 3140–3159.*

[52] S. Litsyn, V. Shevelev, *On ensembles of low-density parity-check codes: asymptotic distance distributions, IEEE Trans. Inform. Theory, 48 (2002), pp. 887–908.*

[53] A. Lubotsky, R. Philips, P. Sarnak, *Ramanujan graphs, Combinatorica, 8 (1988), pp. 261–277.*

[54] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, *Efficient erasure correcting codes, IEEE Trans. Inform. Theory, 47 (2001), pp. 569–584.*

[55] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, *Improved low-density parity-check codes using irregular graphs, IEEE Trans. Inform. Theory, 47 (2001), pp. 585–590.*

[56] C.C. MacDuffee, *The Theory of Matrices,* Chelsea, New York, 1946.

[57] D.J.C. MacKay, *Turbo codes are low-density parity-check codes,* available at `http://www.cs.toronto.edu/∼mackay/abstracts/-turbo-ldpc.html`.

[58] D.J.C. MacKay, *Good error-correcting code based on very sparse matrices, IEEE Trans. Inform. Theory, 43 (1997), pp. 399–431.*

[59] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam, Netherlands, 1977.

[60] B.H. Marcus, R.M. Roth, P.H. Siegel, *Constrained systems and coding for recording channels, pp. 1911-1988,* in *Handbook of Coding Theory,* V.S. Pless and W.C. Huffman (Editors), Elsevier, Amsterdam, Netherlands, 1998.

[61] B.H. Marcus, P.H. Siegel, J.K. Wolf, *Finite-state modulation codes for data storage, IEEE J. Select. Areas Commun., 10 (1992), pp. 5–37.*

[62] G.A. Margulis, *Explicit group theoretical constructions of combinatorial schemes and their applications to the design of expanders and concentrators, Probl. Inform. Transm., 24 (1988), pp. 39–46.*

[63] R.J. McEliece, D.J.C. MacKay, J.-F. Cheng, *Turbo decoding as an instance of Pearl's 'belief propagation'* algorithm, *IEEE J. Select. Areas Commun., 16 (1998), pp. 140–152.*

[64] T. Mittelholzer, *Efficient encoding and minimum distance bounds of Reed-Solomon-type array codes, Proc. Int. Symposium on Inform. Theory (ISIT), 2002, p.282,* Lausanne, Switzerland.

[65] P. Oswald, A. Shokrollahi, *Capacity-achieving sequences for the erasure channel, IEEE Trans. Inform. Theory, 48 (2002), pp. 3017–3028.*

[66] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann, San Francisco, California, 1988.

[67] H. Pfister, I. Sason, R. Urbanke, *Capacity-achieving ensembles for the binary erasure channel with bounded complexity, Proc. IEEE International Symposium on Information Theory (ISIT), June 2004, p.207,* Chicago, Illinois.

[68] O. Reingold, S. Vadhan, A. Wigderson, *Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors, Proc. 41th Annual Symposium on Foundations of Computer Science (FOCS), Nov. 2000, pp. 3–13,* Redondo Beach, California.

[69] T. Richardson, R. Urbanke, *The capacity of low-density parity-check codes under message-passing decoding, IEEE Trans. Inform. Theory, 47 (2001), pp. 599–618.*

[70] T. Richardson, M.A. Shokrollahi, R. Urbanke, *Design of capacity-approaching irregular low-density parity-check codes, IEEE Trans. Inform. Theory, 47 (2001), pp. 619–637.*

[71] T. Richardson, R. Urbanke, *Efficient encoding of low-density parity-check codes, IEEE Trans. Inform. Theory, 47 (2001), pp. 637–656.*

[72] T. Richardson, R. Urbanke, *Modern Coding Theory,* Cambridge University Press, Cambridge, UK, preliminary version, available at `http://lthcwww.epfl.ch/mct`.

[73] R.M. Roth, *Lecture Notes in Coding Theory,* Cambridge University Press, Cambridge, UK, 2006.

[74] R.M. Roth, V. Skachek, *Improved nearly-MDS expander codes, IEEE Trans. Inform. Theory, 52 (2006), pp. 3650–3661.*

[75] R.M. Roth, V. Skachek, *On nearly-MDS expander codes, Proc. IEEE Int. Symposium on Inform. Theory (ISIT), 2004, p. 8,* Chicago, Illinois.

[76] J. Rosenthal, P.O. Vontobel, *Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis, Proc. 38th Annual Allerton Conference on Communication, Control, and Computing, 2000, pp. 248–257,* Monticello, Illinois.

[77] I. Sason, R. Urbanke, *Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel, IEEE Trans. Inform. Theory, 50 (2004), June 2004, pp. 1247–1256.*

[78] C.E. Shannon, *A mathematical theory of communication, Bell System Tech. J. 27 (1948), pp. 379–423 (Part I), pp. 623–656 (Part II) .*

[79] M. Sipser, D.A. Spielman, *Expander codes, IEEE Trans. Inform. Theory, 42 (1996), pp. 1710–1722.*

[80] V. Skachek, R.M. Roth, *Generalized minimum distance iterative decoding of expander codes, Proc. IEEE Inform. Theory Workshop (ITW), Mar.-Apr. 2003, pp. 245–248,* Paris, France.

[81] D.A. Spielman, *Linear-time encodable and decodable error-correcting codes, IEEE Trans. Inform. Theory, 42 (1996), pp. 1723–1731.*

[82] R.M. Tanner, *A recursive approach to low-complexity codes, IEEE Trans. Inform. Theory, 27 (1981), pp. 533–547.*

[83] R.M. Tanner, *Explicit construction of concentrators from generalized N-gons, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 287–293.*

[84] G. Zémor, *On expander codes, IEEE Trans. Inform. Theory, 47 (2001), pp. 835–837.*

[85] V.V. Zyablov, *An estimate of the complexity of constructing binary linear cascaded codes, Problemy Peredachi Informatsii, 15 (1971), pp. 58–70* (in Russian).

Appendix A

We provide here the proofs of Lemmas 2.2.2 and 2.3.2.

Given a bipartite graph $\mathcal{G} = (A : B, E)$, we associate with $\mathcal{G}$ a $|A| \times |B|$ real matrix $X_{\mathcal{G}}$ whose rows and columns are indexed by A and B, respectively, and $(X_{\mathcal{G}})_{u,v} = 1$ if and only if $\{u, v\} \in E$. With a proper ordering on $A \cup B$, the matrix $X_{\mathcal{G}}$ is related to the adjacency matrix of $\mathcal{G}$ by

$$A_{\mathcal{G}} = \left(\begin{array}{c|c} 0 & X_{\mathcal{G}} \\ \hline X_{\mathcal{G}}^T & 0 \end{array} \right) . \tag{A.1}$$

Lemma A.1 *Let $\mathcal{G} = (A : B, E)$ be a bipartite Δ-regular graph where $|A| > 1$. Then Δ^2 is the largest eigenvalue of the (symmetric) matrix $X_{\mathcal{G}}^T X_{\mathcal{G}}$ and the all-one vector $\mathbf{1}$ is a corresponding eigenvector. The second largest eigenvalue of $X_{\mathcal{G}}^T X_{\mathcal{G}}$ is $\gamma_{\mathcal{G}}^2 \Delta^2$.*

Proof. We compute the square of $A_{\mathcal{G}}$,

$$A_{\mathcal{G}}^2 = \left(\begin{array}{c|c} X_{\mathcal{G}} X_{\mathcal{G}}^T & 0 \\ \hline 0 & X_{\mathcal{G}}^T X_{\mathcal{G}} \end{array} \right) ,$$

and recall the following two known facts:

(i) $X_{\mathcal{G}} X_{\mathcal{G}}^T$ and $X_{\mathcal{G}}^T X_{\mathcal{G}}$ have the same set of eigenvalues, each with the same multiplicity [56, Theorem 16.2].

(ii) If λ is an eigenvalue of $A_{\mathcal{G}}$, then so is $-\lambda$, with the same multiplicity [22, Proposition 1.1.4].

We conclude that λ is an eigenvalue of $A_{\mathcal{G}}$ if and only if λ^2 is an eigenvalue $X_{\mathcal{G}}^T X_{\mathcal{G}}$; furthermore, when $\lambda \neq 0$, both these eigenvalues have the same multiplicities in their respective matrices. The result readily follows. □

For real column vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, let $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ be the scalar product $\boldsymbol{x}^T \boldsymbol{y}$ and $\|\boldsymbol{x}\|$ be the norm $\sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$.

Lemma A.2 *Let $\mathcal{G} = (A : B, E)$ be a bipartite Δ-regular graph where $|A| = n > 1$ and let $\boldsymbol{s} = (s_u)_{u \in A}$ and $\boldsymbol{t} = (t_u)_{u \in B}$ be two column vectors in $\mathbb{R}^n$. Denote by σ and τ the averages*

$$\sigma = \frac{1}{n} \sum_{u \in A} s_u \qquad \text{and} \qquad \tau = \frac{1}{n} \sum_{u \in B} t_u \ ,$$

and let the column vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ in $\mathbb{R}^n$ be given by

$$\boldsymbol{y} = \boldsymbol{s} - \sigma \cdot \mathbf{1} \qquad \text{and} \qquad \boldsymbol{z} = \boldsymbol{t} - \tau \cdot \mathbf{1} \ .$$

Define the vector $\boldsymbol{x} \in \mathbb{R}^{2n}$ by

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{t} \end{pmatrix} .$$

Then,

$$|\langle \boldsymbol{x}, A_{\mathcal{G}} \boldsymbol{x} \rangle - 2\sigma\tau\Delta n| \leq 2\gamma_{\mathcal{G}} \Delta \|\boldsymbol{y}\| \cdot \|\boldsymbol{z}\| \ .$$

Proof. First, it is easy to see that $X_{\mathcal{G}} \mathbf{1} = X_{\mathcal{G}}^T \mathbf{1} = \Delta \cdot \mathbf{1}$ and that $\langle \boldsymbol{y}, \mathbf{1} \rangle = \langle \boldsymbol{z}, \mathbf{1} \rangle = 0$; these equalities, in turn, yield the relationship:

$$\langle \boldsymbol{y}, X_{\mathcal{G}} \boldsymbol{z} \rangle = \langle \boldsymbol{s}, X_{\mathcal{G}} \boldsymbol{t} \rangle - \sigma\tau\Delta n \ .$$

Secondly, from (A.1) we get that

$$\langle \boldsymbol{x}, A_{\mathcal{G}} \boldsymbol{x} \rangle = 2 \langle \boldsymbol{s}, X_{\mathcal{G}} \boldsymbol{t} \rangle \ .$$

Hence, the lemma will be proved once we show that

$$|\langle \boldsymbol{y}, X_{\mathcal{G}} \boldsymbol{z} \rangle| \leq \gamma_{\mathcal{G}} \Delta \|\boldsymbol{y}\| \cdot \|\boldsymbol{z}\| \ . \tag{A.2}$$

Let

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$$

be the eigenvalues of $X_{\mathcal{G}}^T X_{\mathcal{G}}$ and let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ be corresponding orthonormal eigenvectors where, by Lemma A.1,

$$\lambda_1 = \Delta^2 \ , \qquad \lambda_2 = \gamma_{\mathcal{G}}^2 \Delta^2 \ , \qquad \text{and} \qquad \boldsymbol{v}_1 = (1/\sqrt{n}) \cdot \mathbf{1} \ .$$

Write

$$\boldsymbol{z} = \sum_{i=1}^{n} \mathsf{B}_i \boldsymbol{v}_i \ ,$$

where $\mathsf{B}_i = \langle \boldsymbol{z}, \boldsymbol{v}_i \rangle$. Recall, however, that $\mathsf{B}_1 = (1/\sqrt{n}) \cdot \langle \boldsymbol{z}, \mathbf{1} \rangle = 0$; so,

$$\begin{aligned} \|X_{\mathcal{G}}\boldsymbol{z}\|^2 &= \langle \boldsymbol{z}, X_{\mathcal{G}}^T X_{\mathcal{G}} \boldsymbol{z} \rangle \\ &= \Big\langle \sum_{i=2}^{n} \mathsf{B}_i \boldsymbol{v}_i, \sum_{i=2}^{n} \lambda_i \mathsf{B}_i \boldsymbol{v}_i \Big\rangle = \sum_{i=2}^{n} \lambda_i \mathsf{B}_i^2 \|\boldsymbol{v}_i\|^2 \\ &\le \lambda_2 \sum_{i=2}^{n} \mathsf{B}_i^2 = \lambda_2 \|\boldsymbol{z}\|^2 = \gamma_{\mathcal{G}}^2 \Delta^2 \|\boldsymbol{z}\|^2 . \end{aligned}$$

The desired result (A.2) is now obtained from the Cauchy-Schwartz inequality. □

Lemma A.3 *Let $\mathcal{G} = (A : B, E)$ be a bipartite Δ-regular graph where $|A| = n > 1$ and let $\chi : (A \cup B) \to \mathbb{R}$ be a function on the vertices of $\mathcal{G}$. Define the function $w : E \to \mathbb{R}$ and the average $\mathsf{E}_{\mathcal{G}}\{w\}$ by*

$$w(e) = \chi(u)\chi(v) \quad \textit{for every edge } e = \{u, v\} \textit{ in } \mathcal{G}$$

and

$$\mathsf{E}_{\mathcal{G}}\{w\} = \frac{1}{\Delta n} \sum_{e \in E} w(e) .$$

Then

$$\Big| \mathsf{E}_{\mathcal{G}}\{w\} - \mathsf{E}_{\mathcal{G}}^A\{\chi\} \cdot \mathsf{E}_{\mathcal{G}}^B\{\chi\} \Big| \le \gamma_{\mathcal{G}} \sqrt{\mathsf{Var}_{\mathcal{G}}^A\{\chi\} \cdot \mathsf{Var}_{\mathcal{G}}^B\{\chi\}} ,$$

where

$$\mathsf{E}_{\mathcal{G}}^A\{\chi^i\} = \frac{1}{n} \sum_{u \in A} (\chi(u)^i) ,$$

$$\mathsf{E}_{\mathcal{G}}^B\{\chi^i\} = \frac{1}{n} \sum_{u \in B} (\chi(u)^i) ,$$

$$\mathsf{Var}_{\mathcal{G}}^A\{\chi\} = \mathsf{E}_{\mathcal{G}}^A\{\chi^2\} - (\mathsf{E}_{\mathcal{G}}^A\{\chi\})^2 ,$$

and

$$\mathsf{Var}_{\mathcal{G}}^B\{\chi\} = \mathsf{E}_{\mathcal{G}}^B\{\chi^2\} - (\mathsf{E}_{\mathcal{G}}^B\{\chi\})^2 .$$

Proof. Define the column vectors

$$\boldsymbol{s} = (\chi(u))_{u \in A} , \quad \boldsymbol{t} = (\chi(u))_{u \in B} ,$$

and

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{t} \end{pmatrix} ,$$

and denote by σ and τ the averages

$$\sigma = \frac{1}{n}\sum_{u\in A} s_u \qquad \text{and} \qquad \tau = \frac{1}{n}\sum_{u\in B} t_u \ .$$

The following equalities are easily verified:

$$\mathsf{E}_{\mathcal{G}}\{w\} = \frac{\langle \boldsymbol{x}, A_{\mathcal{G}}\boldsymbol{x}\rangle}{2\Delta n} \ ,$$

$$\mathsf{E}^{A}_{\mathcal{G}}\{\chi\} = \sigma \ , \qquad \mathsf{E}^{B}_{\mathcal{G}}\{\chi\} = \tau \ ,$$

$$\mathsf{Var}^{A}_{\mathcal{G}}\{\chi\} = \frac{1}{n}\cdot \|\boldsymbol{s} - \sigma\cdot \mathbf{1}\|^2 \ ,$$

and

$$\mathsf{Var}^{B}_{\mathcal{G}}\{\chi\} = \frac{1}{n}\cdot \|\boldsymbol{t} - \tau\cdot \mathbf{1}\|^2 \ .$$

The result now follows from Lemma A.2. □

Proof of Lemma 2.2.2. Using the notation of Lemma A.3, write

$$\mathsf{E}_{\mathcal{G}}\{w\} = \frac{1}{\Delta n}\sum_{u\in A}\sum_{v\in\mathcal{N}(u)} \chi(u)\chi(v) \ , \tag{A.3}$$

$$\mathsf{E}^{A}_{\mathcal{G}}\{\chi\} = \frac{1}{n}\sum_{u\in A}\chi(u) = \sigma \ , \tag{A.4}$$

and

$$\mathsf{E}^{B}_{\mathcal{G}}\{\chi\} = \frac{1}{n}\sum_{u\in B}\chi(u) = \tau \ . \tag{A.5}$$

Since the range of χ is restricted to the interval $[0, 1]$, we have

$$\mathsf{E}^{A}_{\mathcal{G}}\{\chi^2\} \le \mathsf{E}^{A}_{\mathcal{G}}\{\chi\} \qquad \text{and} \qquad \mathsf{E}^{B}_{\mathcal{G}}\{\chi^2\} \le \mathsf{E}^{B}_{\mathcal{G}}\{\chi\} \ ;$$

hence, the values $\mathsf{Var}^{A}_{\mathcal{G}}\{\chi\}$ and $\mathsf{Var}^{B}_{\mathcal{G}}\{\chi\}$ can be bounded from above by

$$\mathsf{Var}^{A}_{\mathcal{G}}\{\chi\} \le \sigma - \sigma^2 \qquad \text{and} \qquad \mathsf{Var}^{B}_{\mathcal{G}}\{\chi\} \le \tau - \tau^2 \ . \tag{A.6}$$

Substituting (A.3)–(A.6) into Lemma A.3 yields

$$\left| \frac{1}{\Delta n}\Big(\sum_{u\in A}\sum_{v\in\mathcal{N}(u)} \chi(u)\chi(v)\Big) - \sigma\tau \right| \le \gamma_{\mathcal{G}}\sqrt{\sigma(1{-}\sigma)\tau(1{-}\tau)} \ ;$$

so,

$$\begin{aligned}
&\frac{1}{\Delta n}\sum_{u\in A}\sum_{v\in\mathcal{N}(u)}\chi(u)\chi(v)\\
&\qquad\leq\ \sigma\tau+\gamma_{\mathcal{G}}\sqrt{\sigma(1-\sigma)\tau(1-\tau)}\\
&\qquad=\ (1-\gamma_{\mathcal{G}})\sigma\tau+\gamma_{\mathcal{G}}\sqrt{\sigma\tau}\left(\sqrt{\sigma\tau}+\sqrt{(1-\sigma)(1-\tau)}\right)\\
&\qquad\leq\ (1-\gamma_{\mathcal{G}})\sigma\tau+\gamma_{\mathcal{G}}\sqrt{\sigma\tau}\ ,
\end{aligned}$$

as claimed. □

Proof of Lemma 2.3.2. We compute lower and upper bounds on the average

$$\frac{1}{\Delta n}\sum_{v\in B}\sum_{u\in\mathcal{N}(v)}\chi(u)\chi(v)\ .$$

On the one hand, this average equals

$$\frac{1}{\Delta n}\sum_{\substack{v\in B:\\ \chi(v)>0}}\chi(v)\underbrace{\sum_{u\in\mathcal{N}(v)}\chi(u)}_{\geq\delta_B\Delta/2}\geq\frac{1}{\Delta n}\cdot\frac{\delta_B\Delta}{2}\underbrace{\sum_{v\in B}\chi(v)}_{\tau n}=\frac{\delta_B\tau}{2}\ ,$$

where the inequality follows from the assumed conditions on χ. On the other hand, this average also equals

$$\frac{1}{\Delta n}\sum_{u\in A}\sum_{v\in\mathcal{N}(u)}\chi(u)\chi(v)\leq(1-\gamma_{\mathcal{G}})\sigma\tau+\gamma_{\mathcal{G}}\sqrt{\sigma\tau}\ ,$$

where the inequality follows from Lemma 2.2.2. Combining these two bounds we get

$$\frac{\delta_B\tau}{2}\leq(1-\gamma_{\mathcal{G}})\sigma\tau+\gamma_{\mathcal{G}}\sqrt{\sigma\tau}\ ,$$

and the result is now obtained by dividing by $\gamma_{\mathcal{G}}\tau$ and re-arranging terms. □

Appendix B

When analyzing the complexity of the algorithm in Figure 2.1, one can notice that the decoder $\mathcal{D} \in \{\mathcal{D}_A, \mathcal{D}_B\}$ needs to be applied at vertex u, only if $(\boldsymbol{z})_{E(u)}$ has been modified since the last application of $\mathcal{D}$ at that vertex. Based on this observation, we prove the following lemma.

Lemma B.1 *The number of (actual) applications of the decoders $\mathcal{D}_A$ and $\mathcal{D}_B$ in the algorithm in Figure 2.1 can be bounded from above by $\omega \cdot n$, where*

$$\omega = 2 \cdot \left\lceil \frac{\log\left(\dfrac{\Delta\beta\sqrt{\sigma}}{\beta - \sigma}\right)}{\log\left(\dfrac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)} \right\rceil + \frac{1 + \dfrac{\delta_A}{\delta_B}}{1 - \left(\dfrac{4\gamma_{\mathcal{G}}^2}{\delta_A\delta_B}\right)^2} .$$

Proof. Define i_T by

$$i_T \;=\; 2 \cdot \left\lceil \frac{\log\left(\dfrac{\Delta\beta\sqrt{\sigma}}{\beta - \sigma}\right)}{\log\left(\dfrac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)} \right\rceil .$$

It is easy to verify that

$$\left(\frac{\delta_A\delta_B}{4\gamma_{\mathcal{G}}^2}\right)^{i_T/2} \left(\frac{1}{\sqrt{\sigma}} - \frac{\sqrt{\sigma}}{\beta}\right) \;\geq\; \Delta \,. \tag{B.1}$$

In the first i_T iterations in Figure 2.1, we apply the decoder $\mathcal{D}$ (which is either $\mathcal{D}_A$ or $\mathcal{D}_B$) at most $i_T \cdot n$ times.

Next, we evaluate the total number of applications of the decoder $\mathcal{D}$ in iterations $i = i_T{+}1, i_T{+}2, \cdots, \nu$. We hereafter use the notations U_i and S_i as in the proof of Theorem 2.3.1. Recall that we need to apply the decoder $\mathcal{D}$ to $(\boldsymbol{z})_{E(u)}$ for a vertex $u \in U_{i+2}$, only if at least one entry in $(\boldsymbol{z})_{E(u)}$ — say, the one that is indexed by the edge $\{u, v\} \in E(u)$ — has been altered during iteration $i+1$. Such an alteration may occur only if v is a vertex in U_{i+1} with an adjacent vertex in S_i. We conclude that $\mathcal{D}$ needs to be applied at vertex u during iteration $i+2$ only if $u \in \mathcal{N}(\mathcal{N}(S_i))$. The number of such vertices u, in turn, is at most $\Delta^2 |S_i| = \Delta^2 \cdot \sigma_i n$.

We now sum the values of $\Delta^2 \sigma_i n$ over iterations $i = i_T+1, i_T+2, \cdots, \nu$:

$$\begin{aligned} \Delta^2 n \cdot \sum_{i=i_T+1}^{\nu} \sigma_i &= \Delta^2 n \left(\sum_{j=i_T/2}^{\lfloor(\nu-1)/2\rfloor} \sigma_{2j+1} + \sum_{j=i_T/2}^{\lfloor(\nu-2)/2\rfloor} \sigma_{2j+2} \right) \\ &\leq \Delta^2 n \cdot \sum_{j=i_T/2}^{\lfloor(\nu-1)/2\rfloor} \sigma_{2j+1} \left(1 + \frac{\delta_A}{\delta_B}\right) , \end{aligned} \tag{B.2}$$

where the last inequality is due to (2.13).

From (2.17) (and by neglecting a positive term), we obtain

$$\frac{1}{\sqrt{\sigma_{i+1}}} \geq \left(\frac{\delta_A \delta_B}{4\gamma_{\mathcal{G}}^2}\right)^{i/2} \left(\frac{1}{\sqrt{\sigma}} - \frac{\sqrt{\sigma}}{\beta}\right)$$

for even $i \geq i_T$. Therefore, the expression in (B.2) is bounded from above by

$$\frac{\Delta^2 n \left(1 + \frac{\delta_A}{\delta_B}\right) \cdot \left(\frac{4\gamma_{\mathcal{G}}^2}{\delta_A \delta_B}\right)^{i_T}}{\left(1 - \left(\frac{4\gamma_{\mathcal{G}}^2}{\delta_A \delta_B}\right)^2\right) \left(\frac{1}{\sqrt{\sigma}} - \frac{\sqrt{\sigma}}{\beta}\right)^2} \leq \frac{\Delta^2 n \left(1 + \frac{\delta_A}{\delta_B}\right) \cdot \frac{1}{\Delta^2}}{1 - \left(\frac{4\gamma_{\mathcal{G}}^2}{\delta_A \delta_B}\right)^2} = \frac{n \left(1 + \frac{\delta_A}{\delta_B}\right)}{1 - \left(\frac{4\gamma_{\mathcal{G}}^2}{\delta_A \delta_B}\right)^2} ,$$

where the inequality follows from (B.1).

Adding now the number of applications of the decoder $\mathcal{D}$ during the first i_T iterations, we conclude that the total number of applications of the decoder $\mathcal{D}$ is at most $\omega \cdot n$, where

$$\omega = i_T + \frac{1 + \frac{\delta_A}{\delta_B}}{1 - \left(\frac{4\gamma_{\mathcal{G}}^2}{\delta_A \delta_B}\right)^2} .$$

□

Appendix C

We provide here the proof of Lemma 3.3.1.

We analyze the error exponent, following the outline of the analysis of Forney [30, Chapter 4.2]. Let ς_i, $i = 1, \cdots, n$, be a random variable which equals 1 if no inner decoding error is made while decoding the i-th inner codeword, and -1 otherwise. The outer code will fail to decode correctly if and only if

$$\varsigma \stackrel{\triangle}{=} \frac{1}{n}\sum_{i=1}^{n} \varsigma_i < (1-2\beta) \ .$$

Denote

$$\mu(-s) \stackrel{\triangle}{=} \ln\left(\mathsf{Prob}_e(\mathcal{C}_{in}) \cdot \mathsf{e}^{s} + (1-\mathsf{Prob}_e(\mathcal{C}_{in})) \cdot \mathsf{e}^{-s}\right) \ .$$

Using the Chernoff bound, we obtain

$$\mathsf{Prob}_e(\mathbb{C}_\Phi) \quad = \quad \mathsf{Prob}\left(\frac{1}{n}\sum_{i=1}^{n} \varsigma_i < (1-2\beta)\right) < \mathsf{e}^{-n(s(2\beta-1)-\mu(-s))} \ .$$

Optimization of the exponent over values of s yields that the maximum of the expression

$$s(2\beta-1)-\mu(-s)$$

is achieved when

$$s = \tfrac{1}{2}\ln\frac{(1-\mathsf{Prob}_e(\mathcal{C}_{in}))\cdot 2\beta}{\mathsf{Prob}_e(\mathcal{C}_{in})\cdot(2-2\beta)} \ ,$$

and the maximum is

$$\begin{aligned} s(2\beta-1)-\mu(-s) \quad = & \quad -\beta\ln\left(\mathsf{Prob}_e(\mathcal{C}_{in})\right) - (1-\beta)\ln\left(1-\mathsf{Prob}_e(\mathcal{C}_{in})\right) \\ + & \quad \beta\ln\left(\beta\right) + (1-\beta)\ln\left(1-\beta\right) \ , \end{aligned}$$

thus completing the proof. □

Appendix D

We provide here the proof of Lemma 3.5.1.

Consider the value of the binary entropy function at the point $p + x$ for small $x > 0$. Using Taylor series around point p,

$$\mathsf{H}_2(p+x) = \mathsf{H}_2(p) + \mathsf{H}_2'(p) \cdot x + \frac{1}{2}\mathsf{H}_2''(p) \cdot x^2 + O(x^3) \ .$$

By calculation of the derivatives of the entropy function, one obtains

$$\begin{aligned} \mathsf{H}_2'(\chi) &= -\log_2 \chi - \chi \cdot \frac{1}{\chi} \cdot \log_2 \mathsf{e} + \log_2(1-\chi) \\ &+ (1-\chi) \cdot \frac{1}{1-\chi} \cdot \log_2 \mathsf{e} = \log_2 \left(\frac{1-\chi}{\chi}\right) \ ; \end{aligned}$$

and

$$\mathsf{H}_2''(\chi) = \log_2 \mathsf{e} \cdot \left(-\frac{1}{1-\chi} - \frac{1}{\chi}\right) = \frac{\log_2 \mathsf{e}}{\chi(\chi-1)} \ .$$

Therefore,

$$\mathsf{H}_2(p+x) = \mathsf{H}_2(p) + \log_2\left(\frac{1-p}{p}\right) \cdot x + \frac{\log_2 \mathsf{e}}{p(p-1)}\frac{x^2}{2} + O(x^3) \ .$$

By applying the inverse of the binary entropy function on both sides of the equation, we get

$$\begin{aligned} p + x &= \mathsf{H}_2^{-1}\left(\mathsf{H}_2(p+x)\right) \\ &= \mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \log_2\left(\frac{1-p}{p}\right) \cdot x + \frac{\log_2 \mathsf{e}}{p(p-1)} \cdot \frac{x^2}{2} + O(x^3)\right) \ . \end{aligned}$$

Denote by θ the value of $\log_2\left(\frac{1-p}{p}\right) \cdot x + \frac{\log_2 \mathsf{e}}{p(p-1)} \cdot \frac{x^2}{2}$, thus obtaining

$$p + x = \mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \theta + O(x^3)\right) \ . \tag{D.1}$$

By solving the quadratic equation

$$\theta = \left(\ln \left(\frac{1-p}{p} \right) \cdot x + \frac{1}{p(p-1)} \cdot \frac{x^2}{2} \right) \cdot \log_2 \mathsf{e} \, ,$$

or equivalently

$$x^2 + 2p(p-1) \ln \left(\frac{1-p}{p} \right) x - \frac{2\theta p(p-1)}{\log_2 \mathsf{e}} = 0 \, ,$$

we obtain two solutions for the intermediate x, namely

$$\begin{aligned} x &= \frac{1}{2} \left(-2p(p-1) \ln \left(\frac{1-p}{p} \right) \pm \sqrt{4p^2(p-1)^2 \ln^2 \left(\frac{1-p}{p} \right) + \frac{8\theta p(p-1)}{\log_2 \mathsf{e}}} \right) \\ &= -p(p-1) \ln \left(\frac{1-p}{p} \right) \pm \sqrt{\left(p(p-1) \ln \left(\frac{1-p}{p} \right) \right)^2 + \frac{2\theta p(p-1)}{\log_2 \mathsf{e}}} \; ; \end{aligned}$$

however, only one of these solutions is positive:

$$x = -p(p-1) \ln \left(\frac{1-p}{p} \right) + \sqrt{\left(p(p-1) \ln \left(\frac{1-p}{p} \right) \right)^2 + \frac{2\theta p(p-1)}{\log_2 \mathsf{e}}} \, .$$

The latter equality can be rewritten as

$$x = p(p-1) \ln \left(\frac{1-p}{p} \right) \cdot \left(-1 + \sqrt{1 + \frac{2\theta}{p(p-1) \left(\ln \left((1-p)/p \right) \right)^2 \log_2 \mathsf{e}}} \right) . \qquad \text{(D.2)}$$

Using Taylor series approximation for small values of χ,

$$\sqrt{1+\chi} = 1 + \frac{1}{2}\chi - \frac{1}{8}\chi^2 + O(\chi^3) \, ,$$

Equality (D.2) becomes

$$\begin{aligned} x &= p(p-1) \ln \left(\frac{1-p}{p} \right) \cdot \Bigg(-1 + 1 + \frac{\theta}{p(p-1) \left(\ln \left((1-p)/p \right) \right)^2 \log_2 \mathsf{e}} \\ &\qquad - \frac{1}{2} \cdot \frac{\theta^2}{p^2(p-1)^2 \left(\ln \left((1-p)/p \right) \right)^4 (\log_2 \mathsf{e})^2} + O(\theta^3) \Bigg) \\ &= \frac{\theta}{\log_2 \left((1-p)/p \right)} - \frac{1}{2} \cdot \frac{\theta^2 \log_2 \mathsf{e}}{p(p-1) \left(\log_2 \left((1-p)/p \right) \right)^3} + O(\theta^3) \, . \qquad \text{(D.3)} \end{aligned}$$

We substitute the evaluation of value of x in (D.3) into Equation (D.1). Thus, we obtain

$$\begin{aligned} \mathsf{H}_2^{-1} \left(\mathsf{H}_2(p) + \theta + O(\theta^3) \right) &= p + \frac{\theta}{\log_2 \left((1-p)/p \right)} \\ &- \frac{1}{2} \cdot \frac{\theta^2 \log_2 \mathsf{e}}{p(p-1) \left(\log_2 \left((1-p)/p \right) \right)^3} + O(\theta^3) \, . \qquad \text{(D.4)} \end{aligned}$$

If $p < \frac{1}{2}$ is fixed and θ is small, then the value of $\mathsf{H}_2(p) + \theta$ is bounded away from 1. In this case, the derivative of $\mathsf{H}_2^{-1}(\chi)$ at point $\chi = \mathsf{H}_2(p) + \theta$ is bounded, and, therefore

$$\mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \theta + O(\theta^3)\right) = \mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \theta\right) + O(\theta^3) \; .$$

Then, the equality (D.4) becomes

$$\mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \theta\right) = p + \frac{\theta}{\log_2\left((1-p)/p\right)} - \frac{1}{2} \cdot \frac{\theta^2 \log_2 \mathsf{e}}{p(p-1)\left(\log_2\left((1-p)/p\right)\right)^3} + O(\theta^3) \; .$$

Finally, we substitute $\theta = \varepsilon(1 - \mathsf{H}_2(p))$ and get that

$$\begin{aligned}\mathsf{H}_2^{-1}\left(\mathsf{H}_2(p) + \varepsilon(1 - \mathsf{H}_2(p))\right) = p + \frac{\varepsilon(1 - \mathsf{H}_2(p))}{\log_2\left((1-p)/p\right)} \\ - \frac{1}{2} \cdot \frac{\varepsilon^2(1 - \mathsf{H}_2(p))^2 \log_2 \mathsf{e}}{p(p-1)\left(\log_2\left((1-p)/p\right)\right)^3} + O(\varepsilon^3) \; ,\end{aligned}$$

thus completing the proof of the lemma. □

Appendix E

We provide here the proofs of Lemmas 4.4.2 and 4.4.4.

Proof of Lemma 4.4.2. Below, we prove equality (4.23) (the proof of equality (4.22) is similar).

We have

$$\begin{aligned}|E_{S\cup T^1}| &= \sum_{v\in T^1} \deg_S(v) \\ &= \sum_{v\in T^1\cap R_\alpha} \deg_S(v) + \sum_{v\in T^1\setminus R_\alpha} \deg_S(v) \\ &= \sum_{v\in T^1\cap R_\alpha} \Delta(\sigma + o_\Delta(1)) + n\Delta \cdot o_\Delta(1) \ ,\end{aligned}$$

where the last transition is due to Lemma 4.4.1. We obtain that

$$\frac{|E_{S\cup T^1}|}{n\Delta} = \sigma\tau_1 + o_\Delta(1) \ . \tag{E.1}$$

Next, we rewrite the definition of Γ, and, using the definition of Γ_v, we obtain

$$|E_{S\cup T^1}| \cdot \Gamma = \sum_{e\in E_{S\cup T^1}} \mathsf{w_b}(e) = \sum_{v\in T^1} \sum_{e\in E_{S\cup\{v\}}} \mathsf{w_b}(e) = \sum_{v\in T^1} \deg_S(v)\cdot\Gamma_v \ . \tag{E.2}$$

The right-hand side of (E.2) can be rewritten as

$$\begin{aligned}\sum_{v\in T^1} \deg_S(v)\Gamma_v &= \sum_{v\in T^1\cap R_\alpha} \deg_S(v)\cdot\Gamma_v + \sum_{v\in T^1\setminus R_\alpha} \deg_S(v)\cdot\Gamma_v \\ &= \sum_{v\in T^1\cap R_\alpha} \Delta\Gamma_v\cdot(\sigma + o_\Delta(1)) + n\Delta\cdot o_\Delta(1) \ . \end{aligned} \tag{E.3}$$

We combine the expressions in (E.1), (E.2) and (E.3) to obtain that

$$\Gamma = \frac{1}{|T^1|} \sum_{v\in T^1\cap R_\alpha} \Gamma_v + o_\Delta(1) \ ,$$

and the claim follows by an application of Lemma 4.4.1. □

Proof of Lemma 4.4.4. Lemma 4.4.2 implies that

$$\sum_{v \in T^1} \Gamma_v \leq (\Gamma + \varepsilon) \cdot |T^1| \ .$$

Consider a subset $T' \subseteq T^1$ of vertices for which $\Gamma_v \leq \Gamma + \varepsilon$. Then, obviously, the ratio $|T'|/|T^1|$ is bounded from below by a constant independent of Δ. On the other hand, the subset $T'' \subseteq T'$ of the vertices that do not satisfy (4.21) (when taking $R_\alpha = T''$) can be made arbitrarily small for small values of α. Therefore, $T' \backslash T'' \neq \emptyset$, and we can pick a vertex $v \in T' \backslash T''$.

Let w_v be a relative q-ary weight of a vector indexed by $E(v)$. Denote by α' the fraction of non-zero edges in $E_{S \cup \{v\}}$. The vertex v was selected such that $\deg_S(v)$ is very close to $\sigma\Delta$, and, therefore, $\alpha' = \mathsf{w}_v/\sigma$ (when ignoring the vanishing terms). From the definitions of B_v and Γ_v, we have

$$\mathsf{B}_v = \Gamma_v/\alpha' \ . \tag{E.4}$$

Lemma 4.4.3 implies

$$\mathsf{w}_v \geq \frac{1 - r_B}{\mathsf{H}_2(\mathsf{B}_v)} \ ,$$

which, in turn, using (E.4) and $\alpha' = \mathsf{w}_v/\sigma$, yields

$$\sigma \geq \frac{1 - r_B}{\alpha' \cdot \mathsf{H}_2(\Gamma_v/\alpha')} \ .$$

Since the binary entropy function is $\cap$-convex, we have

$$\mathsf{H}_2(\Gamma_v) \geq \alpha' \cdot \mathsf{H}_2(\Gamma_v/\alpha')$$

for any value of Γ_v and $\alpha' \leq 1$.

Finally, we use the fact that $\Gamma_v \leq \Gamma$ (while neglecting the ε term) and $\bar{\mathsf{H}}_2(\cdot)$ is nondecreasing, in order to conclude that

$$\mathsf{H}_2(\Gamma_v) \leq \bar{\mathsf{H}}_2(\Gamma) \ ,$$

which proves the claim of the lemma. □

Appendix F

We derive here sufficient conditions on the asymptotic goodness of the code $\mathbb{C}$ defined as in Theorem 6.4.3, and show that the condition in Theorem 6.4.3 is weaker than the presented sufficient conditions. The results in this appendix are formulated for the code having a Δ-*regular* biparite underlying graph $\mathcal{G} = (V = A : B, E)$ (the regularity here means that the value $\deg_{\mathcal{G}}(v)$ is constant for all $v \in V$). However, the results in Lemma F.1 and Theorems F.2, F.3 may be shown for the graphs with $\deg_{\mathcal{G}}(v)$ constant only for the vertices $v \in A$ (rather than $v \in V$). In that case, for vertices $v \in B$ we use (possibly different) constituent codes $\mathcal{C}_B(v)$ of length $|E(v)|$ having minimum distance greater than or equal d_B.

Lemma F.1 *Let $\mathcal{G}$ be a bipartite Δ-regular (α, ζ)-expander. Let $\mathbb{C}$ be the code defined as in Section 2.1, with the linear codes $\mathcal{C}_A$ and $\mathcal{C}_B$ over $\mathbb{F}$ of minimum distances $d_A = \delta_A \Delta$ and d_B, respectively. Let $\mathbf{c} = (c_e)_{e \in E}$ be a non-zero codeword in $\mathbb{C}$ with a support Y (namely, $Y = \{e \in E \ : \ c_e \neq 0\}$), and let σn and τn be the sizes of subsets of vertices in A and B, respectively, that are endpoints of the edges in Y. Then, for $\sigma \leq \alpha$,*

$$(1 - \zeta)\sigma\Delta \geq \tau(d_B - 1) \ , \tag{F.1}$$

and for $\tau \leq \alpha$,

$$(1 - \zeta)\tau\Delta \geq \sigma(d_A - 1) \ . \tag{F.2}$$

Proof. Let $S \subseteq A$ be the set of vertices that are endpoints of edges in Y. Then $|S| = \sigma n$. Suppose that $\sigma \leq \alpha$. We define T, T_0 and T_1 as in the proof of Theorem 6.4.3, namely $T \subseteq B$ is the set of neighbors of vertices in S. The sets T_0 and T_1 are defined as

$$T_0 = \{v \in T \ : \ c_e = 0 \text{ for all } e \in E(v)\} \ ,$$

and

$$T_1 = T \backslash T_0 \ .$$

Note that $T_1 = \tau n$.

The total number of edges in the graph $\mathcal{G}_{S \cup T}$ (the graph induced by the vertex set $S \cup T$) is $\sigma \Delta n$. Each edge in $E_{S \cup T}$ (the edge set of the graph $\mathcal{G}_{S \cup T}$) is incident with a vertex either in T_0 or in T_1.

Edges incident with vertices in T_1. Each vertex in T_1 has at least d_B non-zero edges in $E_{S \cup T}$ incident with it (in addition to possible zero edges in $E_{S \cup T}$). Therefore, the number of such edges is at least

$$d_B \cdot \tau n \ . \tag{F.3}$$

Edges incident with vertices in T_0. Due to the expansion property, the number of vertices in T is at least $\zeta \Delta \cdot \sigma n$. The number of vertices in T_1 is τn. Therefore, the number of vertices in T_0 is at least

$$\zeta \Delta \cdot \sigma n - \tau n \ . \tag{F.4}$$

For each vertex in T_0, there is at least one edge in $E_{S \cup T}$ incident with it. Therefore, the number of edges incident with vertices in T_0 is bounded from below by the expression (F.4).

We sum the sizes of the sets of these two types of edges incident with T_0 and T_1 to obtain the lower bound on the number of edges in $E_{S \cup T}$. This yields

$$\sigma \Delta n \geq d_B \tau n + \zeta \Delta \sigma n - \tau n \ .$$

After some simplifications, we obtain (F.1), as required. The inequality (F.2) is obtained similarly, by switching between the sets A and B. □

Theorem F.2 *Let $\mathbb{C}$ be the code defined as in Lemma F.1, with the linear codes $\mathcal{C}_A$ and $\mathcal{C}_B$ over $\mathbb{F}$ of minimum distance $d_A = d_B$. Let $\mathcal{G}$ be a bipartite (α, ζ)-expander. If*

$$(1 - \zeta)\Delta < d_B - 1 \ , \tag{F.5}$$

then the relative minimum distance of $\mathbb{C}$ is bounded from below by $\alpha d_A / \Delta$.

Proof. Let $\boldsymbol{c} \in \mathbb{C}$, σ and τ be defined as in Lemma F.1. Suppose (w.l.o.g.) that $\sigma \leq \tau$. We assume (to the contrary) that $\sigma \leq \alpha$. Observe that the conditions of Lemma F.1 are satisfied, and, therefore, the inequality (F.1) holds. Then, from (F.1), using $\sigma \leq \tau$, we obtain

$$(1 - \zeta)\Delta \geq d_B - 1 \ , \tag{F.6}$$

thereby reaching a contradiction.

Recall that $d_A = d_B$. In case $\sigma > \tau$, we obtain the same expression (F.6) by using (F.2).

Finally, we have that if condition (F.5) holds, then $\min\{\sigma, \tau\} > \alpha$, and, therefore, the relative minimum distance of $\mathbb{C}$ is bounded from below by $\alpha d_A / \Delta$. □

Theorem F.3 *Let $\mathbb{C}$ be the code defined as in Lemma F.1, with the linear codes $\mathcal{C}_A$ and $\mathcal{C}_B$ over $\mathbb{F}$ of minimum distances d_A and d_B, respectively. Let $\mathcal{G}$ be a bipartite (α, ζ)-expander. If*

$$(1-\zeta)^2\Delta^2 < (d_A - 1)(d_B - 1) , \tag{F.7}$$

then the relative minimum distance of $\mathbb{C}$ is bounded from below by $\min\{\alpha d_A/\Delta, \alpha d_B/\Delta\}$.

Proof. Let $\boldsymbol{c} \in \mathbb{C}$, σ and τ be defined as in Lemma F.1. We assume (to the contrary) that $\sigma \leq \alpha$ and $\tau \leq \alpha$. Observe that the conditions of Lemma F.1 are satisfied, and, therefore, the inequalities (F.1) and (F.2) are true. Then, we multiply (F.1) and (F.2), and reduce the resulting inequality by the factor $\sigma\tau$, yielding

$$(1-\zeta)^2\Delta^2 \geq (d_A - 1)(d_B - 1) . \tag{F.8}$$

As before, we obtain that if the condition (F.7) holds, then $\max\{\sigma, \tau\} > \alpha$, and, therefore, the relative minimum distance of $\mathbb{C}$ is bounded from below by either $\alpha d_A/\Delta$, or $\alpha d_B/\Delta$, thus completing the proof. □

In the next example we compare the condition for asymptotic goodness of $\mathbb{C}$ in Theorem F.3 with its counterpart in Theorem 6.4.3.

Example F.1 Take δ_A very close to (but smaller than) $\frac{1}{2}$, set $d_B = 3$ and $\zeta = \frac{3}{4}$. Then, the condition (6.7) is satisfied, thus ensuring that the overall code $\mathbb{C}$ is asymptotically good.

On the other hand, the condition (F.7) is not satisfied for big values of Δ, and therefore it cannot be used to show the asymptotic goodness of $\mathbb{C}$.

It can be shown that if the condition in Theorem F.3 is satisfied for some values of ζ, d_A and d_B, then the condition in Theorem 6.4.3 is satisfied for these values. Indeed, rewrite condition (F.7) as

$$(1-\zeta)^2\Delta^2 < (d_A - 1)(d_B - 1) = \left(\frac{d_B}{d_A}(d_A - 1)\right) \cdot \left(\frac{d_A}{d_B}(d_B - 1)\right) .$$

Therefore, either

$$(1-\zeta)\Delta < \frac{d_A}{d_B}(d_B - 1) \tag{F.9}$$

or

$$(1-\zeta)\Delta < \frac{d_B}{d_A}(d_A - 1) . \tag{F.10}$$

The inequality (F.9) can be re-written as

$$d_B(1-\zeta)\Delta < d_A(d_B - 1) ,$$

and

$$d_A < d_A d_B - d_B(1-\zeta)\Delta = d_B(d_A - (1-\zeta)\Delta) \ .$$

The latter inequality is equivalent to inequality (6.7) — this can be seen by dividing both sides by $d_A - (1-\zeta)\Delta$.

If the inequality (F.10) holds, then similarly, by switching between d_A and d_B, it can be re-written in the form (6.7).

Conclusion. Theorem 6.4.3 (possibly, with switching between $\mathcal{C}_A$ and $\mathcal{C}_B$) yields Theorem F.3.