Title

Advances in Protograph-Based LDPC Codes and a Rate Allocation Problem

Author

Vasista Srinivasan Ranganathan, Sudarsan

Publication Date

2018

UNIVERSITY OF CALIFORNIA

Los Angeles

Advances in Protograph-Based LDPC Codes and a Rate Allocation Problem

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical and Computer Engineering

by

Sudarsan Vasista Srinivasan Ranganathan

2018

ABSTRACT OF THE DISSERTATION

Advances in Protograph-Based LDPC Codes and
a Rate Allocation Problem

by

Sudarsan Vasista Srinivasan Ranganathan
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2018
Professor Richard D. Wesel, Chair

This dissertation consists of three parts. The first part focuses on a class of modern channel codes known as protograph-based low-density parity-check (LDPC) codes. Also known as protograph LDPC codes, these powerful error-correcting codes have enabled communication systems of the past fifteen years to achieve very high throughputs. The first part of the dissertation presents a new design method based on an upper bound on minimum distance to obtain rate-compatible, protograph quasi-cyclic (QC) LDPC codes called Protograph-based Raptor-like LDPC codes (PBRL codes). A major contribution here is a very-low-complexity PBRL design algorithm that is provably efficient.

The second part of the dissertation continues the focus on protograph LDPC codes, first exploring how the decoding complexity of PBRL codes can be reduced and whether the extending structure that provides rate-compatibility to a PBRL code is optimal or not. Then, this part considers the problem of design of PBRL codes for any increment ordering. The degree-1 extending structure yields naturally to the design of PBRL codes that decode efficiently even when increments arrive out-of-order. This part finally considers the following question: What is the shortest block-length required to obtain a protograph QC-LDPC code with a girth of at least 6 or 8 from a $(3, L)$ complete protograph? An affirmative answer is given for girth of at least 6 and directions are explored for girth of at least 8.

Finally, the dissertation turns to communication theory and tackles a rate allocation problem previously studied in literature, but with an important twist. Consider a cross-layer coding scheme with packet-level erasure coding and physical-layer channel coding. It is known from previous work that some erasure coding is necessary even in the limit of large physical-layer codeword block-lengths if the physical-layer fading channel does not provide diversity that grows with block-length. However, is erasure coding still required in the limit of large block-lengths if the physical layer allows for diversity to grow with block-length? The theoretical answer turns out to be a resounding “no” in the case of Rayleigh fading that allows diversity to increase linearly with block-length.

The dissertation of Sudarsan Vasista Srinivasan Ranganathan is approved.

Suhas N. Diggavi

Christina Panagio Fragouli

Songwu Lu

Dariush Divsalar

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2018

To Vatsala, Neela, Vasu, and Vasiya.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This dissertation would not have been possible without the support of my family, teachers, mentors, colleagues, and friends. First and foremost, I want to thank my family: Amma, Appa, Paati, and Vasu have always cheered me on and have always encouraged me to pursue what I liked the most. Paati, who passed away to cancer in 2013 and whom I miss sorely, is one of my role models. She will be a source of inspiration for me during hard times because of how she fought her battle against cancer and how she never gave up till her very last day. I am forever indebted to her unconditional love. I am where I am today because of Amma and Appa. They have toiled hard to raise me and Vasu to achieve what we wanted to do in life. For that, I know I am blessed with wonderful parents and I will always be grateful. Vasu, my sister, is someone I admire for her smart work and I look forward to a lifetime of supporting each other (and quarreling with each other) as family.

I am grateful to my wonderful advisors at UCLA – Professors Richard Wesel (Rick) and Dariush Divsalar. Rick's passion and interest in a wide variety of problems have rubbed off on me, and I am thankful to his generous support, thoughtful guidance, and unlimited patience throughout my years in the Communication Systems Laboratory (CSL). Rick's energy to work on every research problem and his mentorship have enabled me to pursue even the smallest of research ideas. His attention to identifying the impacts of a result and their implications are unmatched. Rick's interest in problems that have a real-world impact is something I hope I can emulate. The dissertation would not have been possible without Dariush's guidance and commitment to research problems that we worked on together. His attention to even the most basic ideas, his encouragement towards pursuing those ideas, and his dedication towards research have been the major enablers of my research. He provided me tools and taught me techniques that made my graduate research journey a fruitful one. Rick and Dariush, I hope to continue collaborating on research problems with you.

As for a few other professors at UCLA that I am grateful for – Professor Lara Dolecek's

support and encouragement through the CoDESS initiative was a key part of my early research at UCLA. I want to thank Professors Lieven Vandenberghe, Suhas Diggavi, Songwu Lu, Greg Pottie, Monica Visan, Michael Hitrik, Raghu Meka, and Jonathan Kao (and Rick Wesel, of course) for teaching courses that have played a crucial role in the development of my thinking. I also want to thank Professors Pottie, Diggavi, and Vandenberghe, in particular, for letting me be a teaching assistant for their courses on Introduction to Electrical Engineering, Communication Systems, and Mathematics of Design, respectively. These teaching experiences taught me things beyond what coursework can teach.

Elsewhere, I want to thank BZ Shen for his supervision and mentorship during my two summer internship experiences at Broadcom Corporation. The research on problem in Chapter 4 was largely due to my time at Broadcom. BZ let me pursue research the way I wished to and his full support during my time at Broadcom was vital to my exponential rate of learning I experienced during my times there.

My time at UCLA was possible because of staff members whose services I have benefited from while being at UCLA. I wish to thank Deeona Columbia, Mandy Smith, Michelle Welch, Ryo Arreola, Sandra Bryant, Myrna Reneau, and Jose Cano for all the assistance and help they have provided to me.

I have benefited from the wonderful colleagues I have had at UCLA. I am honored to have worked with Kasra Vakilinia, Haobo Wang, Will Chuang, Nathan Wong, Hengjie Yang, Tong Mu, and many others. Kasra mentored me meticulously and helped me sail through my first few years in the CSL research group. I am indebted to all of his work that I have built upon for my research. I wish I had worked on a few more research problems with Haobo, and I want to thank him for all the conversations we have shared and all the trips around Los Angeles that we took for fun. Haobo was also a roommate for the last two years of my time at UCLA and he is the best roommate I have ever had. I wish the junior graduate students Will Chuang, Hengjie Yang, Linfang Wang, and Ethan Liang the best of luck over their graduate careers.

Life during my graduate research years would have been very hard without the wonderful friends I have always had. I want to thank Manu anna, Vidhyashri, Karthick, Krithika, Nisha, and Disha for a lifetime of experiences. Manu anna, Vidhyashri, Karthick, and Nisha, in particular, have been wonderful companions that I have always counted upon. I cherish all the times that we spent together in Los Angeles, Santa Barbara, Bay Area, and national parks. Atharav and Vigneshwar, two of my roommates, have been a source of long conversations about engineering, research, and other things; I will remember our times. To my friends who have been there for me for a long time – VP, Prithivi, Swamy, Venky, Yamunai, Prasad, Kesari, Moni, Vigneshwari, Suba, Pavi, Prasy, and Swats – I do not have to tell you how much I value your friendships.

And finally, to the love of my life – Vasiya Krishnan. I thank you for your support and patience. Your love and patience know no limits and I am excited to start our journey through life as a married couple.

Almost surely, I have forgotten to include people in this list that deserve to be here. Please forgive the oversight.

The work in this dissertation was possible, in large part, due to the computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Institute for Digital Research and Education's Research Technology Group. I am grateful to the generous funding support that I have received from National Science Foundation (NSF), Broadcom Foundation, and S. A. Photonics during various stages of my research.

Vita

2012	B.E. (Electronics and Communication Engineering) SSN College of Engineering, Anna University, Chennai, India
2013	Chairman's Medal SSN College of Engineering
2013–2018	Graduate Student Researcher Electrical & Computer Engineering Department UCLA, Los Angeles, California, USA
2013–2018	Teaching Assistant (elevated to Teaching Associate and Teaching Fellow) Electrical & Computer Engineering Department, UCLA
2014	M.S. (Electrical Engineering) UCLA
2014	Tech Intern (Coding Theory) Broadcom Corporation, Irvine, California, USA
2015	Outstanding Poster Award, 2015 IEEE Comm. Theory Workshop Dana Point, California
2015	Tech Intern (Coding Theory) Broadcom Corporation
2015–2016	Secretary Engineering Graduate Students Association (eGSA), UCLA
2016–2017	President Engineering Graduate Students Association (eGSA), UCLA

Selected Publications

S. V. S. Ranganathan, D. Divsalar, K. Vakilinia, and R. D. Wesel, "Design of high-rate irregular non-binary LDPC codes using algorithmic stopping-set cancellation," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2014.

S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "On the girth of (3,L) quasi-cyclic LDPC codes based on complete protographs," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2015.

S. V. S. Ranganathan, K. Vakilinia, L. Dolecek, D. Divsalar, and R. D. Wesel, "Some results on spatially coupled protograph LDPC codes," in *Inform. Theory Applicat. Workshop (ITA)*, 2016.

K. Vakilinia, **S. V. S. Ranganathan**, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary LDPC examples," *IEEE Trans. Commun.*, 2016.

S. V. S. Ranganathan, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Universal rate-compatible LDPC code families for any increment ordering," in *Int. Symp. Turbo Codes & Iterative Inform. Processing (ISTC)*, 2016.

S. V. S. Ranganathan, T. Mu, and R. D. Wesel, "Allocating redundancy between erasure coding and channel coding when fading channel diversity grows with codeword length," *IEEE Trans. Commun.*, 2017.

S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Design of improved quasi-cyclic protograph-based raptor-like LDPC codes for short blocklengths," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2017.

H. Wang, **S. V. S. Ranganathan**, and R. D. Wesel, "Approaching capacity using incremental redundancy without feedback," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2017.

S. V. S. Ranganathan, R. D. Wesel, and D. Divsalar, "Linear rate-compatible codes with degree-1 extending variable nodes under iterative decoding," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2018.

S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths," Under revision, *IEEE Trans. Inform. Theory*, 2018.

CHAPTER 1

Quasi-Cyclic Protograph-Based Raptor-Like LDPC Codes for Short Block-Lengths

Protograph-based Raptor-like low-density parity-check codes (PBRL codes) [CVDW15] are a family of easily encodable rate-compatible LDPC codes. PBRL codes have an excellent performance across all design rates. Quasi-cyclic (QC) [Bla03] PBRL code families permit high-speed decoder implementations. PBRL codes designed thus far, for both long and short block-lengths, have been based on optimizing the iterative decoding threshold of the protograph of the PBRL family at various design rates.

This chapter[1] introduces a design method to obtain better QC PBRL code families at short block-lengths (of a few hundred bits) for low frame error rate (FER) requirements. We first select a protomatrix for the highest design rate. To add a new row to lower the rate, we keep all the previously obtained rows of the PBRL protomatrix fixed and select the new row that maximizes an upper bound on the minimum distance of any QC-LDPC code that can be obtained from the protomatrix. The new QC PBRL code families outperform the original PBRL codes at short block-lengths by providing a significantly better low-FER performance.

The chapter uses the new design method to produce new short block-length PBRL codes designed for parameters in the 5G wireless standard. The new codes perform significantly

[1] Parts of this chapter were presented at the 2017 IEEE International Symposium on Information Theory (ISIT) [RDW17]. Almost all of the material in this chapter has been submitted as a journal paper (under revision) to IEEE Transactions on Information Theory [RDW18].

better in the low-FER regime than the proposed PBRL codes in the 5G wireless standard.

1.1 Introduction

Rate-compatible (RC) channel codes are at the core of systems with incremental redundancy (IR). IR at the physical layer of a system allows efficient transmission over a time-varying channel. RC codes operate in phases: First, the transmitter of the system sends the highest-rate codeword of a certain size over the channel. If the receiver is unable to decode the received noisy codeword, the transmitter then sends additional codeword symbols to the receiver to allow the receiver attempt another decoding round, but now with a longer received codeword. The process continues until either a maximum number of transmissions have been attempted or the decoder succeeds in decoding the received codeword.

RC channel codes that use soft decisions at the receiver were first introduced by Hagenauer in [Hag88]. Here, the author proposed using a punctured convolutional code (See Cain et al. [CCG79].) as the highest-rate code, with the transmitter sending symbols that are initially punctured as IR. These codes are called RC punctured convolutional (RCPC) codes. More recently, RC punctured turbo (RCPT) codes, introduced by Narayanan and Stüber in [NS97], have been demonstrated to have excellent throughput performances. RCPC codes and RCPT codes are schemes that are both based on punctured convolutional codes.

This chapter focuses on RC codes based on low-density parity-check (LDPC) codes, which were introduced by Gallager in [Gal63]. Optimized LDPC code ensembles have iterative decoding thresholds very close to channel capacity over various discrete memoryless channels (See Richardson and Urbanke [RU01] and [RU08].) As demonstrated by Richardson et al. in [RSU01], despite possessing a sub-optimal iterative decoder, irregular LDPC codes designed according to optimized degree distributions of the underlying Tanner graph [Tan81] have frame error rate (FER) performances very close to capacity at sufficiently long block-lengths. Their excellent performance comes at a cost, though, as decoders of unstructured LDPC codes have a high implementation complexity. Therefore, LDPC codes with some structure in their parity-check matrix are generally preferred in practice.

Protograph-based LDPC codes are one class of structured LDPC codes. A protograph is a small Tanner graph that can be used to obtain LDPC codes of various block-lengths with the same structural connections as the protograph. First described by Thorpe in [Tho03], LDPC codes based on protographs are amenable to tractable analysis and design procedures. It is possible to obtain the iterative decoding threshold of a protograph LDPC code ensemble using only the protograph, as shown by Divsalar et al. [DDJA09], Liva and Chiani [LC07], and others. The ensemble weight enumerators of a protograph can also be obtained, as shown by Abu-Surra et al. [ASDR11] and others.

Protograph-based quasi-cyclic LDPC (QC-LDPC) codes [Fos04], a class of protograph codes, have parity-check matrices composed of circulant permutation matrices (CPMs) and permit very-low-complexity decoder implementations [DDJA09]. The presence of CPMs in protograph QC-LDPC codes facilitates analysis relating the connections in the protograph to the girth and minimum distance of the corresponding code. Fossorier [Fos04], Karimi and Banihashemi [KB13], and others analyze the girth of a protograph QC-LDPC code by examining the protograph. More pertinent to this chapter are [SV12] and [BS13]. Smarandache and Vontobel [SV12] derive an upper bound on the minimum distance of any QC-LDPC code that can be obtained from a protograph. Butler and Siegel [BS13] extend the results of [SV12] to QC-LDPC codes based on punctured protographs.

1.1.1 Rate-Compatible LDPC (RC-LDPC) Codes: Prior Work

Rate-compatible LDPC (RC-LDPC) codes are generally designed via one of the two following approaches. The first approach constructs the lowest-rate code and then punctures symbols to obtain corresponding codes of higher rates. The second approach, called extending, constructs the highest-rate code and then constructs additional symbol as combinations of existing symbols to obtain the lower-rate codes.

Some notable works that start with a low-rate code and use the puncturing approach include [HKM04, HKKM06, KRM09, EKHB09, VF09]. Although these works construct RC-LDPC code families with good performance, Yazdani and Banihashemi [YB04] have observed that finite-length RC-LDPC code families obtained solely using puncturing a low-rate code do not perform competitively at relatively high rates. Further, compared with the objectives of our chapter to design practical codes with high-throughput decoders, Ha et al. [HKM04], Ha et al. [HKKM06], Kim et al. [KRM09], and Vellambi and Fekri [VF09] do not explicitly design structured RC-LDPC code families with high-throughput decoders. El-Khamy et al. [EKHB09] construct codes based on protographs, but they start with a low-rate code and use puncturing.

This chapter constructs RC code families using the extending method. Works that have explored this approach previously include [YB04, Dol05, NND12, NN13, CVDW15]. Yazdani and Banihashemi [YB04] show that a combination of both extending and puncturing can improve over puncturing approaches. Dolinar [Dol05], Nguyen et al. [NND12], Nguyen and Nosratinia [NN13], and Chen, Vakilinia et al. [CVDW15] all construct QC protograph-based RC-LDPC code families.

This chapter builds upon the work of Chen, Vakilinia et al. in [CVDW15]. In [CVDW15], the authors introduce protograph-based Raptor-like LDPC codes (PBRL codes). PBRL code families are a class of easily encodable RC-LDPC codes. If constructed using circulants, QC PBRL codes also lead to high-throughput decoder implementations. PBRL code families of [CVDW15], despite possessing a simple structure, have an excellent iterative decoding threshold and performance across all design rates. In [CVDW15], the authors design PBRL protographs for long and short block-lengths by optimizing the iterative decoding threshold of the protograph at each successively lower rate. They show that PBRL QC-LDPC code families can outperform other QC RC-LDPC codes in the literature (including the ones of [NND12] and [NN13]), both at short (≈ 1000 information bits) and long (≈ 16000 information bits) block-lengths.

1.1.2 Contributions

This chapter considers the design of structured, i.e. quasi-cyclic, RC-LDPC codes for very short block-lengths (fewer than 500 information bits) and for low FER requirements. The previous literature mentioned in Section 1.1.1 use iterative decoding threshold as the design metric in their RC-LDPC code design. While the iterative decoding threshold is an effective design metric for predicting the waterfall performance at long block-lengths [RU01], it is not a good indicator of short block-length performance or performance generally at low FERs. In particular, a good threshold does not preclude low-weight codewords, stopping sets [DPT$^+$02, TJVW04], trapping sets [Ric03], and absorbing sets [ZDN$^+$06], which can lead to poor waterfall performance for short block-length codes and high error floors generally. This chapter proposes the use of an upper bound on the minimum distance as a potential metric for short block-length RC-LDPC code design.

Consider how an upper bound on minimum distance $d_{\min}$ relates to the size of the smallest codewords, stopping sets, trapping sets, and absorbing sets. An upper bound on $d_{\min}$ is an upper bound on the size (weight) of the smallest weight codeword for any linear code including LDPC codes. Consider the graph induced by the variable nodes corresponding to a codeword, which we refer to as the graph induced by the codeword. An upper bound on $d_{\min}$ is also an upper bound on the size of the smallest stopping set because the variable nodes of the graph induced by any codeword form a stopping set [TJVW04].

An (a, b) trapping set is a set of a variable nodes with $b \geq 0$ odd-degree check nodes in the sub-graph induced by the a variable nodes. Note that the induced graph of a weight-$d_{\min}$ codeword has zero odd-degree check nodes and is therefore a trivial trapping set with $a = d_{\min}$ and $b = 0$. Removing a single variable node and its edges from the induced graph of the codeword generates a trapping set with $a = d_{\min} - 1$ and b equal to the degree of the removed variable node. Thus for each variable node v with degree d_v present in the graph induced by the smallest-weight codeword, there exists an $a = d_{\min} - 1$, $b = d_v$ trapping set so that $d_{\min} - 1$ is an upper bound on the minimum size of $b = d_v$ trapping sets. Thus, the $d_{\min}$ of the code provides an upper bound on the size of numerous trapping sets, and when $d_{\min}$ is small, these will affect the error floor performance.

An (a, b) absorbing set is an (a, b) trapping set where each variable node has strictly more even degree check nodes than odd degree check nodes as neighbors in the induced sub-graph. Similar to trapping sets, codewords are trivial absorbing sets and removing any degree-d_v variable node from the induced graph of the codeword generates a trapping set that is also an absorbing set whenever d_v is less than half the number of check nodes in the induced graph of the codeword.

As discussed above, the variable nodes of the graph induced by any codeword form a trivial stopping set, trapping set, and absorbing set. Thus $d_{\min}$ is an upper bound on the size of the smallest of each of these sets. Moreover, we have shown how nontrivial trapping sets and absorbing sets of size $d_{\min} - 1$ are natural consequences of the minimum-weight codeword.

Still, there are limitations to the analysis above. Indeed, $d_{\min} - 1$ may not be an upper bound on the size a of a trapping set or absorbing set with a other values of b. Furthermore, maximizing an upper bound on a quantity does not guarantee maximizing the quantity itself. However, we conjecture and support with results that show that selecting the rows of a PBRL incremental redundancy matrix to maximize an upper bound on $d_{\min}$ leads to a low-complexity design procedure that produces short block-length PBRL codes with better low-FER performance than existing design techniques of Section 1.1.1, which make no attempt at all to directly address low-FER performance of short block-length LDPC codes.

Our contributions are as follows:

Permanent-Bound-Based Design Method

One contribution of this chapter is a new PBRL design approach. Given a set of design rates, we design protographs for PBRL ensembles by maximizing, at each rate, the upper bounds on the minimum distance that were derived in [SV12] and [BS13]. The resulting PBRL QC-LDPC code families outperform the ones designed by optimizing the iterative decoding threshold by providing a better low-FER performance slope at each design rate. The new approach has the advantage that it is oblivious to the channel that the RC code is to be deployed, which is not the case with the method of optimizing the iterative decoding threshold.

We note that the upper bounds derived in [SV12] and [BS13] have been used previously, for example, by Mitchell et al. [MSC14] to design QC-LDPC block codes based on small protographs.

Reduction in Complexity of Computing the Upper Bounds for PBRL Protomatrices

The complexity of evaluating the upper bounds of [SV12] and [BS13], in general, increases exponentially with the size of the overall protograph. A second contribution of this chapter is to leverage the structure of PBRL protographs to identify a significant reduction in this complexity. In particular, we show in Theorems 3, 4, and 5 and Lemma 3 that the structure of a PBRL protomatrix, given a few conditions that are satisfied for most designs, facilitates computation of the upper bounds with complexity that grows only linearly with the size of the overall PBRL protomatrix when the size of the highest-rate code protomatrix is fixed. Further, this reduction in complexity yields a very low-complexity PBRL search procedure that has a constant run-time for each row added to lower the design rate. The constant complexity is in contrast to the threshold-based optimization proposed in [CVDW15] where the complexity of the PBRL search procedure grows with each new added row.

Equivalence Between the Design Method and an Integer Linear Program (ILP)

The results on complexity reduction lead to an equivalent way of representing the design of a row of the PBRL protomatrix. We show that the exhaustive search for a new row that maximizes the upper bounds of [SV12] or [BS13] is equivalent to an integer linear program (ILP).

1.1.3 Organization

The chapter is organized as follows: Section 1.2 introduces notation and background. Section 1.3 presents the permanent-bound-based PBRL design method. Section 1.4 shows that the structure of a PBRL protomatrix allows for a significant reduction in the complexity of computing the permanent-based upper bounds. Based on the results of Section 1.4, Section 1.5 establishes the equivalence between the new design procedure for a row of a PBRL protomatrix and an integer linear program (ILP). Section 1.6 provides example PBRL protomatrices designed for short block-lengths according to the new design method. Section 1.7 presents the results obtained from simulations of the new QC PBRL code families and compares the results against PBRL codes designed by optimizing the iterative decoding threshold.

1.2 Preliminaries and Notation

1.2.1 LDPC Codes, Protographs, and PBRL Codes

LDPC Codes and Tanner Graphs

An LDPC code [Gal63] is a linear code [Bla03] with a sparse parity-check matrix. Associated with any parity-check matrix is what is called the *Tanner graph* of the code. A Tanner graph is a bipartite graph whose two sets of nodes represent the rows and columns, respectively, of a parity-check matrix. The nodes representing the rows are called *check nodes* and the nodes representing the columns are called *variable nodes*. Two nodes (a variable-check node pair) are connected in a Tanner graph by an edge whenever the entry located at the corresponding row/column is non-zero. The *degree* of a node in a Tanner graph is the number of edges emanating from the node. The *girth* of a graph is the length of the shortest cycle present in the graph.

Protographs

A *protograph* [Tho03], or a projected graph, is a small bipartite Tanner graph with n_c check nodes and n_v variable nodes forming the two node sets. The biadjacency matrix of a protograph is called its *protomatrix*. In this chapter, we use the terms protograph and protomatrix interchangeably. The protomatrix can contain entries that are greater than 1. These entries correspond to multiple edges between the same variable-check node pair in the protograph.

Protograph-Based LDPC Codes

Given a protograph, we define a two-step *copy-and-permute* process, also known as *lifting*, to obtain an LDPC code as follows. The protograph is first replicated M times, yielding M disconnected copies of the graph. Then, the connections to the check nodes of the set of M edges (across the M replicas) obtained from an edge of a variable-check node pair in the original protograph are permuted among the M copies of the corresponding variable-check node pair. The same is performed independently for every set of M edges. The second step of lifting connects the M copies of the protograph such that the resulting LDPC code has the same degree distribution as the protograph. The resulting LDPC code has Mn_c check nodes and Mn_v variable nodes; M is called the *lifting factor*.

Of the n_v variable nodes in a protograph, some of the nodes can be designated to be "punctured". The number of such punctured nodes is denoted n_p and the remaining $n_t = n_v - n_p$ nodes are called "transmitted" nodes. The *design rate* of a protograph is $R = \frac{n_v - n_c}{n_t}$. Upon lifting a protograph, all the nodes that are copied from a punctured variable node in the protograph are punctured in the resulting graph and all the nodes that are copied from a transmitted variable node in the protograph are transmitted in the resulting graph. The rate r of a protograph-based LDPC code obtained from a protograph with design rate R satisfies $r \geq R$. We call a protograph (protomatrix) a punctured protograph (punctured protomatrix) if it contains punctured variable nodes and call it an unpunctured protograph (unpunctured protomatrix) otherwise.

With the lifting factor being M, the set of all LDPC codes obtainable via all possible edge permutations (in the second step of lifting) from a protograph is collectively called the *ensemble* of a protograph.

A *quasi-cyclic (QC)* protograph LDPC code is obtained from a protograph by restricting the permutations used during the lifting process to a sub-class known as *circulants.* Circulant permutations lead to LDPC codes with parity-check matrices that have blocks of circulant permutation matrices (CPMs) and zero matrices (if there are 0's in the protomatrix) of size $M \times M$.

PBRL Codes

A *protograph-based Raptor-like (PBRL) ensemble* is defined by its protomatrix P, which has the following general form:

$$P = \begin{bmatrix} P_{\mathsf{HRC}} & 0 \\ P_{\mathsf{IRC}} & I \end{bmatrix}_{n_c \times n_v} \tag{1.1}$$

Here, 0 and I refer to the all-zeros and identity matrices of appropriate size. The highest-rate code (HRC) of the rate-compatible protomatrix is represented by P_{HRC}, which is of size $n_{c_H} \times n_{v_H}$. We refer to this sub-matrix as the "HRC part". The variable nodes of the protomatrix containing the identity matrix in (1.1) represent the incremental redundancy symbols of P. We denote the sub-matrix of the incremental redundancy variable nodes by P_{IR}. The check nodes that connect to the degree-1 variable nodes in P_{IR} have connections to the variable nodes in the HRC part. These connections are represented by the sub-matrix P_{IRC}, which we refer to as the "IRC part".

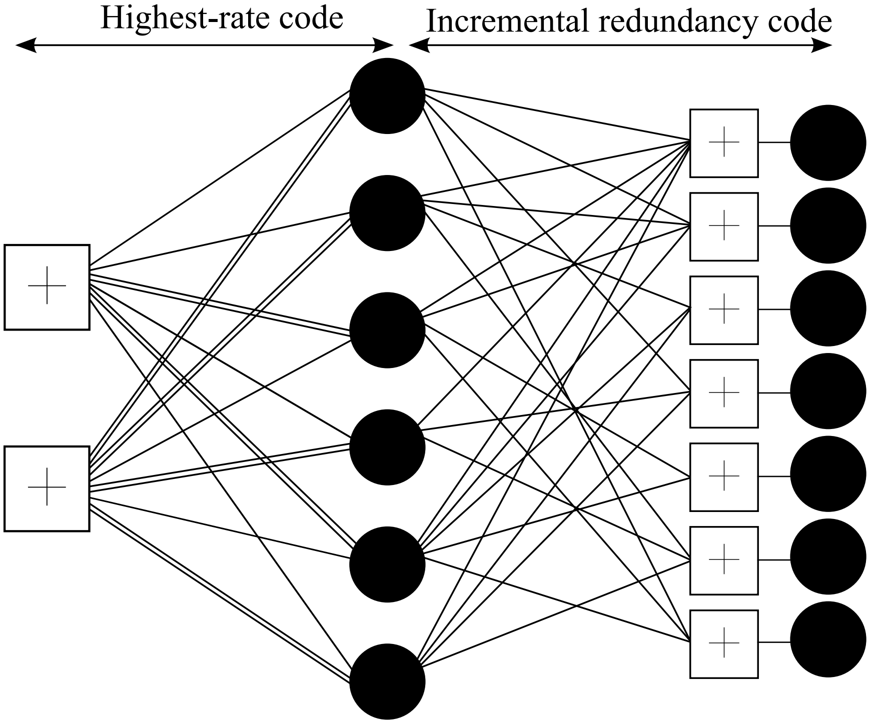

Figure 1.1: A PBRL protograph (no punctured nodes) with a highest-rate code of design rate 2/3 and its incremental redundancy nodes from sub-matrix P_{IR}. The sub-matrix P_{IR} lowers the design rate as its degree-1 variable nodes are included one at a time. The figure is a reproduction of Fig. 1 of [CVDW15]. The protomatrix corresponding to this protograph is given in (1.2).

The HRC part of the PBRL protograph, upon lifting, is structurally identical to the precode part of a Raptor code of Shokrollahi [Sho06]. Similarly, the degree-1 variable nodes of P_{IR} are efficiently encoded as modulo-2 sums of the precode symbols in a manner similar to the Luby transform (LT) code in a Raptor code.

Our convention throughout this dissertation is that the n_p punctured variable nodes of a PBRL protomatrix are present in the first n_{v_H} columns of the PBRL protomatrix.

Example 1. *Fig. 1.1 shows an example PBRL protograph, with black circles representing variable nodes and squares with an XOR symbol representing check nodes. Rate-compatibility starts with the highest-rate code. P_{IR} lowers the design rate as its degree-1 variable nodes are* *included one at a time. The protomatrix corresponding to the protograph of Fig. 1.1 is*

$$P = \begin{bmatrix} 1\,1\,2\,1\,2\,1\,0\,0\,0\,0\,0\,0\,0 \\ 2\,2\,1\,2\,1\,2\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0 \\ 1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix} \tag{1.2}$$

The HRC *part has a design rate of* $4/6$ *and the seven IR variable nodes lead to the lower design rates of* $4/(6+i)$*. We refer the reader to Section II of [CVDW15] for a more detailed introduction to features of PBRL codes.*

Other Notation and Assumptions

In this chapter, we denote the set of integers $\{1, 2, \ldots, \ell\}$ by $[\ell]$. The cardinality of a set S is represented by $|S|$. The indexing of rows and columns of a matrix starts from 1. Given a vector x, x_i denotes the scalar value at position $i \geq 1$ of x. If P is a protomatrix of size $n_c \times n_v$ and $S \subseteq [n_v]$, denote by P_S the sub-matrix of P formed by the columns indexed by elements of S. For convenience, we always assume the columns of P that are indexed by S to appear in P_S in the same order that they appear in P. If $i \in S$, we let $S \setminus i$ be shorthand for $S \setminus \{i\}$. For a set of non-negative integers, let $\min^*$ be a function that returns the smallest non-zero value if the set has at least one positive value or return $+\infty$ if the set is $\{0\}$. In this chapter, all unpunctured protomatrices of size $n_c \times n_v$ are assumed to have a positive design rate, i.e. $n_v > n_c$. Likewise, we assume that all punctured protomatrices of size $n_c \times n_v$ with n_p punctured variable nodes have a positive design rate less than 1, i.e.

$n_v > n_c$, $n_v - n_c < n_v - n_p$. If P is a PBRL protomatrix of size $n_c \times n_v$ (punctured or unpunctured) with an HRC part of size $n_{c_H} \times n_{v_H}$, we assume that P_{HRC} also has a positive design rate less than 1. Therefore, for a punctured PBRL protomatrix $n_p < n_{c_H} \leq n_c$. For simplicity, we call the permanent of a square matrix of size $\ell \times \ell$ a "permanent of size $\ell \times \ell$".

1.2.2 Bounding $d_{\min}$ of Protograph QC-LDPC Codes

A key feature of QC-LDPC codes based on protomatrices is that the minimum distance of any such code obtained from a protomatrix is upper bounded by a constant that depends only on the protomatrix and not on the block-length of the code. In order to state the upper bounds, which were derived in [SV12] and [BS13], we need to define the permanent of a square matrix.

Definition 1 (Permanent). *The permanent of an $\ell \times \ell$ square matrix A with elements $\mathsf{A}(i,j), i \in [\ell], j \in [\ell]$ over some commutative ring is defined as*

$$\mathsf{perm}(\mathsf{A}) = \sum_{\sigma} \prod_{j=1}^{\ell} \mathsf{A}(j, \sigma(j)) = \sum_{\sigma} \prod_{j=1}^{\ell} \mathsf{A}(\sigma(j), j), \tag{1.3}$$

where σ refers to a permutation of $[\ell]$ and the summation is over all permutations of $[\ell]$. The permanent, although it looks deceptively similar to the determinant, is harder to compute than the determinant [Von13]. While the arithmetic complexity of computing the determinant is $O\left(\ell^3\right)$, the most efficient algorithm known to compute the permanent of any square matrix, due to Ryser [Rys63], is of complexity $\Theta\left(\ell \cdot 2^{\ell}\right)$.

Theorem 1 (Upper bound for unpunctured protomatrices; Theorem 8 of Smarandache and Vontobel [SV12]). *Let P be an unpunctured protomatrix of size $n_c \times n_v$. Any QC-LDPC code $\mathcal{C}$ obtained from the protomatrix P has a minimum distance $d_{min}(\mathcal{C})$ that is upper bounded as*

$$d_{min}(\mathcal{C}) \leq \underset{S \subseteq [n_v], |S| = n_c + 1}{min^*} \sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right). \tag{1.4}$$

Note that permanents computed from sub-matrices of a protomatrix are always non-negative.

Theorem 2 (Upper bound for punctured protomatrices; Theorem 9 of Butler and Siegel [BS13])**.** *Let P be a protomatrix of size $n_c \times n_v$. Let the index set of punctured variable nodes, a subset of $[n_v]$, be denoted $\mathcal{P}$. Denote by $\mathcal{C}'$ any punctured QC-LDPC code that can be obtained from P using $\mathcal{P}$, and denote the unpunctured version of the code $\mathcal{C}'$ (with protograph P) by $\mathcal{C}$. Then, provided that $\mathcal{C}$ and $\mathcal{C}'$ have the same dimensionality, i.e. the same number of codewords in their codebooks, $\mathcal{C}'$ has a minimum distance $d_{min}(\mathcal{C}')$ that is upper bounded as*

$$d_{min}(\mathcal{C}') \leq \underset{S \subseteq [n_v], |S| = n_c + 1}{min^*} \sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right). \tag{1.5}$$

1.3 Permanent-Bound-Based PBRL Design

The design of a PBRL protomatrix, as proposed by Chen, Vakilinia et al. in [CVDW15], consists of two steps: First, we choose the HRC part, P_{HRC}, as a protomatrix by itself. Then, we obtain the IRC part, P_{IRC}, one row at a time. In [CVDW15], the authors first choose an HRC part with a degree distribution and an acceptable iterative decoding threshold. They then design each row of P_{IRC} successively to optimize the iterative decoding threshold of the PBRL protomatrix at the rate induced by adding that row, keeping all previously obtained rows fixed. The best performing families of high-throughput RC-LDPC codes at both short and long block-lengths are the PBRL codes as designed with the heuristics proposed by Chen, Vakilinia et al. in [CVDW15].

While the work in [CVDW15] used iterative decoding threshold as the design metric at both short and long block-lengths, it is well known in literature on LDPC codes that threshold is not indicative of how a finite block-length code performs in the low-FER regime. Further, at very short block-lengths the waterfall region of an LDPC code may not yield low FERs such as 10^{-6}. The authors in [CVDW15] proposed constraints on the connections in P_{IRC} in their threshold-based designs to obtain a better slope in low-FER performance at block-lengths such as 1000 information bits and longer, but their work does not address even shorter block-lengths such as fewer than 500 information bits. In this chapter, we use the upper bounds of Theorems 1 and 2, which are applicable for QC-LDPC codes based on protographs, to design PBRL protomatrices for short block-length and practical RC-LDPC codes for low FERs.

Minimum distance, by itself, does not fully predict the low-FER performance of LDPC codes under iterative decoding. However, if one LDPC code has a better graphical structure and minimum distance properties compared to another one that has a similar or better threshold and both codes have the same block-length, then it has been observed in prior works such as [MSC14] that the former usually performs better in terms of the slope of FER performance at low FERs once the channel parameter improves beyond the initial waterfall region. The channel parameter at which the code with the better graphical structure and minimum distance properties performs better than the one with the similar or better threshold will depend upon the difference in their thresholds as threshold indicates the channel parameter at which an LDPC code can start decoding correctly, regardless of its block-length. We show through examples in this chapter that designing QC RC-LDPC codes using the minimum distance upper bounds without taking into consideration the threshold yields better low-FER performance at very short block-lengths when compared to code families designed solely based on optimizing the threshold.

With Theorems 1 and 2 in hand, we propose the following PBRL design procedure that replaces iterative decoding threshold from [CVDW15] with the upper bounds on the minimum distance as the design criterion for every row of P_{IRC}:

1. HRC selection: Choose an HRC matrix of size $n_{c_H} \times n_{v_H}$ with a desired degree distribution and complexity constraint. A common complexity constraint is to limit the weight of each column and row in the protomatrix. Note that the selection of a desirable HRC part is a code design problem by itself. In order to avoid high error floors, one usu-

ally needs to limit the number of degree-2 variable nodes in the protomatrix. Design examples in Sections 1.6 and 1.7 provide HRC parts that lead to good performance in both waterfall and error floor regions.

2. IRC design: Select the next row of the protomatrix from a set of candidate rows to maximize the upper bound on the minimum distance via Theorem 2 or 1 (depending upon whether there are punctured variable nodes or not). If there are multiple candidates with the best upper bound, then select one at random.

3. Go to Step 2) if another row of P_{IRC} is required. Otherwise, exit.

It is not known, in general, whether the upper bounds of (1.4) or (1.5) are achievable. But our design procedure yields better QC RC-LDPC code families at short block-lengths than the design based on optimizing the iterative decoding thresholds.

For a punctured protomatrix, care must be taken to ensure that not too many variable nodes are punctured. Otherwise, the dimensionality requirement in Theorem 2 may be violated.

The structure of PBRL protomatrices and the upper bounds of (1.4) and (1.5) lead to the following observation that will be utilized in the rest of the chapter:

Lemma 1. *Let a PBRL protomatrix P be of size $n_c \times n_v$. Let the* HRC *part, as a protomatrix by itself, have a positive and finite upper bound d_{HRC}, computed using* (1.4) *or* (1.5)*. Then the upper bounds for each new row $i = n_{c_H}+1, n_{c_H}+2, \ldots, n_c$ added to obtain P, irrespective of the chosen candidates for the rows, are non-decreasing and are lower bounded by d_{HRC}.*

Proof. Let us consider the design of the first row of the IRC part and assume that the protomatrix has no punctured columns. Assume that there is no non-zero integer in the new $(n_{c_H}+1)^{\text{th}}$ row of the overall protomatrix except the required 1 from sub-matrix P_{IR} at entry $(n_{c_H}+1, n_{v_H}+1)$ of P. For any set S of $n_{c_H}+2$ column indices that does not include index $n_{v_H}+1$, the sum of the $n_{c_H}+2$ permanents, computed according to the upper bound of (1.4), is zero. For any other set that includes index $n_{v_H} + 1$, the sum of the permanents is equal to one of the sums of $n_{c_H} + 1$ permanents computed to find the upper bound for the HRC part. Now, if the new row is designed to have non-zero entries in the first n_{v_H} columns, the upper bound can only increase or remain the same.

Similar arguments follow if the protomatrix has punctured columns. This completes the proof as the above arguments can then be successively applied to each new row. □

We end the section by discussing the complexity of computing the upper bound of (1.4) for any general protomatrix of size $n_c \times n_v$ with no punctured columns (similar arguments follow if the protomatrix has punctured columns). The number of sets S that need to be considered is $\binom{n_v}{n_c+1}$. For every such set we need to compute $n_c + 1$ permanents, each of size $n_c \times n_c$. Therefore, the overall number of permanents that need to be computed in order to obtain the upper bound of (1.4) is $\binom{n_v}{n_c+1} \cdot (n_c + 1)$. As the complexity of computing the permanent of a square matrix of size $\ell \times \ell$ is $\Theta\left(\ell \cdot 2^{\ell}\right)$, the design will eventually become prohibitively slow as we add more rows to the PBRL protomatrix. In the next section we show how we can take advantage of the general structure of a PBRL protomatrix to significantly reduce the complexity of computing the upper bounds, thereby reducing the complexity of our design procedure.

1.4 Reduced-Complexity Design Procedure

In Sections 1.4.1, 1.4.2, and 1.4.3 below, for a given PBRL protomatrix, we leverage the general structure of the protomatrix of a PBRL ensemble in (1.1) to reduce the complexity of computing the upper bounds in (1.4) or (1.5). Additionally, in Section 1.4.4 below, we reduce the complexity of the design procedure by using computations completed for all the previous rows to facilitate very-low-complexity computation of the bound while designing the next row.

1.4.1 Unpunctured PBRL Protographs

This subsection deals with the case when the PBRL protomatrix has no punctured variable nodes. The following result leads to a significant reduction in both the number of permanents that need to be computed and the size of each permanent to be computed while computing the upper bound of (1.4) for an unpunctured PBRL protomatrix.

Theorem 3. *Let an unpunctured PBRL protomatrix P be of size $n_c \times n_v$. Assume that the upper bound in* (1.4) *for P is a positive integer (this is satisfied whenever the* HRC *part has an upper bound that is a positive integer, as a result of Lemma 1). Then, the same upper bound can be obtained by computing exactly $\binom{n_{v_H}}{n_{c_H}+1} \cdot (n_c + 1)$ permanents, each of complexity at most the complexity of computing the permanent of an $(n_{c_H} + 1) \times (n_{c_H} + 1)$ matrix.*

Before we provide the proof, we comment on the reduction in complexity of computing (1.4). The complexity of computing each permanent would, crucially, now depend only on the number of check nodes in the HRC part, n_{c_H}, and not on n_c. Also, the dominating factor in the expression for number of permanents to be computed is the binomial coefficient, which again would now depend only on the size of the HRC part, $n_{c_H} \times n_{v_H}$, and not on the size of the entire protomatrix. As a result of this theorem, the complexity of computing the upper bound for an unpunctured PBRL protomatrix grows only linearly with the size of the protomatrix $n_c \times n_v$. We remark that such a reduction in complexity will be a recurring theme throughout this section.

Proof. We provide the proof in three cases. The proof is an argument that if P has a positive and a finite upper bound (1.4), then due to the identity matrix structure in the columns of P_{IR} in a PBRL protomatrix it suffices to consider sets S that always contain the indices of all the columns of P_{IR}.

Case 1

Let us first consider the case when $S \subseteq [n_v]$, $|S| = n_c + 1$, and S contains the indices of the last $n_v - n_{v_H}$ columns, i.e. of the columns in P_{IR}. Note that $n_v - n_{v_H} = n_c - n_{c_H}$. There are $\binom{n_{v_H}}{n_{c_H}+1}$ such sets. The $n_c + 1$ chosen columns indexed by any such set S form a matrix with structure that can be written as:

$$P_S = \left[c_1 \ c_2 \ \cdots \ c_{n_{c_H}+1} \mid P_{\mathsf{IR}}\right]_{(n_c \times (n_c+1))} \tag{1.6}$$

where the first $n_{c_H} + 1$ columns are chosen from the initial n_{v_H} columns of P and P_{IR} has the following structure:

$$P_{\mathsf{IR}} = \begin{bmatrix} 0 \\ \\ I \end{bmatrix}_{\left(n_c \times \left(n_v - n_{v_H}\right)\right)} \tag{1.7}$$

Because each column in P_{IR} contains only a single 1, when the removed column is from the P_{IR} part of P_S the complexity of computing the required permanent $\mathsf{perm}\left(P_{S\backslash i}\right)$ is the complexity of computing the permanent of an $(n_{c_H} + 1) \times (n_{c_H} + 1)$ sub-matrix.

When the removed column index $i \in S$ is not an element of $\{n_{v_H} + 1, n_{v_H} + 2, \ldots, n_v\}$, the complexity of computing the permanent of the remainder square matrix $P_{S\backslash i}$ is the complexity of computing the permanent of an $n_{c_H} \times n_{c_H}$ sub-matrix. This is because the product $\prod_{j=1}^{n_c} P_{S\backslash i}(\sigma(j), j)$ is zero for permutations σ that select elements not in the first n_{c_H} rows for the first n_{c_H} columns of $P_{S\backslash i}$.

Case 2

Now let us consider the general set S of $n_c + 1$ column indices in $[n_v]$. First, let us assume that

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) > 0, \tag{1.8}$$

which implies that at least one of the $n_c + 1$ permanents is positive. Denote by $P_{i^*} \triangleq P_{S \setminus i^*}$ one such $n_c \times n_c$ sub-matrix of P_S with a positive permanent. Assume the following definition of a permanent of a square matrix A of size $n_c \times n_c$:

$$\mathsf{perm}\,(A) = \sum_{\sigma} \prod_{j=1}^{n_c} A(\sigma(j), j) \tag{1.9}$$

There exists a permutation denoted σ^* that has a positive product in (1.9) when computed for the matrix P_{i^*}. Consider all columns indexed by $j \in [n_c]$ such that $\sigma^*(j) > n_{c_H}$. There are $n_c - n_{c_H} = n_v - n_{v_H}$ such columns. Replace all these columns of P_{i^*} by the columns of the sub-matrix P_{IR} (whenever possible), in the following manner: Replace column j whose $\sigma^*(j) > n_{c_H}$ with the column in P_{IR} whose only non-zero element, 1, is present in row $\sigma^*(j)$, unless the column from P_{IR} is already in the $n_c + 1$ columns indexed by set S under consideration. Denote by P_1 the newly obtained matrix of size $n_c \times n_c$. P_1 has a permanent that is positive and is at most the value of the permanent of P_{i^*} due to the following reasons: Permutation σ^* yields a positive product with P_1 because the replacements (whenever possible) only lead to non-zero entries at locations $(\sigma^*(j), j) : \sigma^*(j) > n_{c_H}$. Furthermore, each permutation σ that yields a positive product $\prod P_{i^*}(\sigma(j), j)$ in P_{i^*} (including σ^*) yields a product with P_1 that is upper bounded by the product computed with P_{i^*}.

Let us denote by $P_{\hat{S}}$ the matrix $\left[\hat{c}_1 \ \hat{c}_2 \ \cdots \ \hat{c}_{n_{c_H}+1} \mid P_{\mathsf{IR}}\right]$, where $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{n_{c_H}+1}$ are the columns in P_S chosen from the first n_{v_H} columns of the protomatrix P and that either were in P_{i^*} and not replaced to obtain P_1 or was the one column of P_S not in P_{i^*}. It is now straight-forward to see from the composition of the matrices P_S and $P_{\hat{S}}$ that

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) \geq \sum_{i \in \hat{S}} \mathsf{perm}\left(P_{\hat{S} \setminus i}\right) > 0. \tag{1.10}$$

Case 3

We now consider the final case of the general set of $n_c + 1$ columns whose $n_c + 1$ permanents sum to zero. Recall that the statement of the theorem assumes that P has a positive upper bound in (1.4). Therefore, we may ignore such a case unless these columns contain P_{IR}, in which case we would compute the sum of $n_c + 1$ permanents according to the complexity as shown already in Case 1.

This completes the proof as the three cases show it suffices to consider subsets $S \subseteq [n_v]$ that always contain the indices of all the columns of P_{IR}. □

1.4.2 PBRL Protographs with One Punctured Variable Node

Theorem 3 provides a considerable reduction in the complexity of the new design procedure to obtain unpunctured PBRL protomatrices. We now consider PBRL protomatrices with exactly one punctured variable node. First, we start by providing an example that shows that the proof technique of replacing columns followed in Theorem 3 for the general set S of $n_c + 1$ column indices may not necessarily be directly applicable for PBRL protomatrices with punctured variable nodes if S contains the index of a punctured column.

Example 2. *Let us consider the following PBRL protomatrix whose* HRC *part is of size* $n_{c_H} = 2$ *and* $n_{v_H} = 6$ *with the first column punctured:*

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 1 & 0 \\ 0 & 2 & 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.11}$$

Let $S = \{1, 2, 3, 4\}$, the set of indices of the first four columns of the protomatrix. The sum $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right)$ from (1.5) *for this set has three terms in it and is equal to 17. The permanent of the 3×3 sub-matrix comprised of columns $1, 3,$ and 4 is equal to 5. If we follow the same replacement strategy as in Theorem 3, we need to replace column 1 by the only incremental redundancy variable node, i.e. column 7. The new set of column indices $\hat{S} = \{2, 3, 4, 7\}$ has no column that is punctured and the summation $\sum_{i \in \hat{S} \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \setminus i}\right)$, which now has four terms, is equal to 19.*

We show in the following result that, by dealing with the set S that contains the index of exactly one punctured variable node more carefully, the complexity of computing the upper bound for protomatrices with a single punctured column can be reduced by a magnitude similar to Theorem 3.

Theorem 4. *Let a PBRL protomatrix P be of size $n_c \times n_v$ with $n_p = 1$ punctured column. Without loss of generality assume that $\mathcal{P} = \{1\}$. Assume that there exists at least one non-zero entry in the first n_{c_H} rows of all of the first n_{v_H} columns of P (this is a requirement that will be satisfied in any practical design since otherwise there will be an all-zeros column in the* HRC *part, but we are stating this explicitly as we will use this assumption in the proof). Assume that the upper bound in* (1.5) *for P is a positive integer. Then, the same upper bound can be obtained with at most $\binom{n_{v_H}}{n_{c_H}+1} \cdot (n_c + 1)$ permanents, each of size at most $(n_{c_H} + 1) \times (n_{c_H} + 1)$.*

Proof. If $S \subseteq [n_v]$, $|S| = n_c + 1$, $S \cap \mathcal{P} = \phi$, and S does not contain the indices of all the columns of P_{IR}, then the arguments in Theorem 3 for ignoring such a set S while computing the upper bound for the protomatrix still hold. Similarly, if S includes the indices of all the columns of P_{IR} and any subset of columns from the first n_{v_H} columns of P, then the computational complexity arguments for computing the sum of at most $n_c + 1$ permanents for such a set of columns hold the same way as observed in Theorem 3.

Now consider any set $S \subseteq [n_v]$, $|S| = n_c + 1$, and $S \cap \mathcal{P} \neq \phi$. Similar to Theorem 3, assume that

$$\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) > 0 \tag{1.12}$$

as otherwise such a set S can be ignored (unless it contains the indices of all the columns of P_{IR}) because of the assumption that the upper bound for the protomatrix is a positive integer. This implies that there exists an $n_c \times n_c$ sub-matrix $P_{S \setminus i}$ with the punctured column whose permanent is non-zero. Note that every permanent in the summation in (1.12) includes the punctured column. In the rest of the proof, the permanent of a matrix A is assumed to be computed with the equation $\mathsf{perm}\left(\mathsf{A}\right) = \sum_{\sigma} \prod \mathsf{A}(\sigma(j), j)$. We now prove the theorem through the following three cases:

Case 1

Consider the case when there exists a sub-matrix $P_{S \setminus i}$, $i \in S \setminus \mathcal{P}$ with a non-zero permanent wherein there exists a permutation σ^* that yields a non-zero product in the computation of the permanent such that $\sigma^*(1) \leq n_{c_H}$. For this case, since the non-zero element from the punctured column contributing to the non-zero product for this permutation is chosen from the first n_{c_H} rows of the sub-matrix, the replacement strategy of Theorem 3 can be employed to obtain a new set $\hat{S}$ containing the indices of the punctured column and all the columns of P_{IR} such that $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) \geq \sum_{i \in \hat{S} \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \setminus i}\right) > 0$. In particular, both $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right)$ and $\sum_{i \in \hat{S} \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \setminus i}\right)$ have the same number of terms.

Case 2

Consider the $n_c \times n_c$ sub-matrix $P_{S \setminus 1}$, the sub-matrix without the punctured column. This sub-matrix is not part of the summation in (1.12). But, if this sub-matrix has a non-zero permanent (assumption for this case), then for a permutation σ^* whose product is non-zero in $P_{S \setminus 1}$ there are n_{c_H} columns whose contributing non-zero elements to the product are from the first n_{c_H} rows. Since we have assumed that there exists at least one non-zero entry in the first n_{c_H} rows of the punctured column, there exists a column j in $P_{S\setminus 1}$ with $\sigma^*(j) \leq n_{c_H}$ that can be replaced by the punctured column and the product for the permutation σ^* for this new sub-matrix, which is one of $P_{S\setminus i}$ for some $i \in S \setminus \mathcal{P}$, is still non-zero. Now, this implies we arrive at Case 1 with the new sub-matrix and therefore the replacement strategy follows for this new sub-matrix that is included in the summation in (1.12). Note that this shows Case 2 implies Case 1.

Case 3

For the last case, assume the complement of the union of Cases 1 and 2 for the set S. Denote the n_c+1 columns by c_1, c_2, $\ldots$, c_{n_c+1} with c_1 denoting the punctured column with index 1 in the original protomatrix. Consider a sub-matrix P_1 that has a non-zero permanent in (1.12). According to the assumptions for this case, any permutation that has a non-zero product $\prod_{j=1}^{n_c} P_1(\sigma(j), j)$ in this sub-matrix has $\sigma(1) > n_{c_H}$. Pick a permutation σ^* that has a non-zero product with P_1. Replace each column $j \neq 1$ of this sub-matrix whose $\sigma^*(j) > n_{c_H}$ by the column of P_{IR} whose only non-zero element, 1, is in row $\sigma^*(j)$ if the latter is already not indexed in S. Call the new $n_c \times n_c$ sub-matrix P_1'. This sub-matrix has a positive permanent that is upper-bounded by the permanent of P_1. Denote by $\hat{S}$ the index set of $n_c + 1$ columns that lead to the n_c columns of P_1' and the one column indexed in S and was not part of the sub-matrix P_1. It follows that

$$\sum_{i \in S\setminus\mathcal{P}} \mathsf{perm}\left(P_{S\setminus i}\right) \geq \sum_{i \in \hat{S}\setminus\mathcal{P}} \mathsf{perm}\left(P_{\hat{S}\setminus i}\right) > 0. \tag{1.13}$$

Now, if the column indexed in S that was not part of P_1 is the column from P_{IR} whose only non-zero element is in row $\sigma^*(1)$, then the proof is complete as $\hat{S}$ would then have the indices of all the columns of P_{IR}. If this is not the case, then we can replace the index of the punctured column in $\hat{S}$ with the index of the column in P_{IR} whose only non-zero element is in row $\sigma^*(1)$ to obtain a set $\hat{S}_1$ (without the index of the punctured column) that satisfies

$$\sum_{i \in \hat{S} \backslash \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \backslash i}\right) \geq \sum_{i \in \hat{S}_1} \mathsf{perm}\left(P_{\hat{S}_1 \backslash i}\right) > 0. \tag{1.14}$$

This is because of the assumption of complement of Case 2. That is, for the set of columns indexed by S, the permanent of the sub-matrix not containing the punctured column is assumed to be zero. This implies that for $\hat{S}$ the permanent of the sub-matrix not containing the punctured column is still zero. Therefore, for $\hat{S}_1$ the permanent is zero for the sub-matrix not containing the column from P_{IR} that has its only non-zero element in row $\sigma^*(1)$. Thus the proof is complete. □

We note that in Example 2 the set $S = \{1, 2, 3, 4\}$ belongs to Case 1 of the proof. For the sub-matrix indexed by columns $1, 2,$ and 3, which has a non-zero permanent, there exists a σ^* with a non-zero product such that $\sigma^*(1) \leq n_{c_H} = 2$.

1.4.3 Protographs with More than One Punctured Variable Node

We now consider PBRL protographs that have more than one punctured variable node. We first provide an example that shows that the assumptions in Theorem 4 adapted to more than one punctured column are not sufficient to make the replacement steps followed so far for a set S to provide a set $\hat{S}$ with the indices of all the columns of P_{IR} such that $\sum_{i \in S \backslash \mathcal{P}} \mathsf{perm}\left(P_{S \backslash i}\right) \geq \sum_{i \in \hat{S} \backslash \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \backslash i}\right) > 0$.

Example 3. *Consider the following protomatrix with its first two columns punctured (i.e.*

$\mathcal{P} = \{1, 2\}$*):*

$$P = \begin{bmatrix} 1 & 1 & 5 & 5 & 5 & 5 & 0 & 0 \\ 1 & 1 & 5 & 5 & 5 & 5 & 0 & 0 \\ 1 & 1 & 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.15}$$

Let $S = \{1, 2, 3, 4, 5, 7\}$*;* $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) = 900$*. The column index from the incremental redundancy part that is missing in S is 8. According to the replacement steps followed in proofs so far, in order to replace one of the column indices in S to obtain* $\hat{S}$ *that contains both 7 and 8, either of columns 1 or 2 must be replaced as columns 3, 4, and 5 do not have a non-zero integer in row 5. Since columns 1 and 2 are identical, assume without loss of generality that* $\hat{S} = \{1, 3, 4, 5, 7, 8\}$*. Now,* $\sum_{i \in \hat{S} \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S} \setminus i}\right) = 1200$*, which is larger than 900.*

The following theorem provides one set of sufficient conditions under which the replacement strategy used to prove Theorems 3 and 4 is successful when a PBRL protograph has more than one punctured variable node.

Theorem 5. *Let a PBRL protomatrix* P *of size* $n_c \times n_v$ *have* $n_p > 1$ *punctured variable nodes. Without loss of generality assume that* $\mathcal{P} = \{1, 2, \ldots, n_p\}$*. Assume that every entry in the first* n_{c_H} *rows of all the punctured columns is non-zero (this is a stricter requirement than assumed in Theorem 4) and that there exists at least one non-zero entry in the first* n_{c_H} *rows of every one of the unpunctured columns in the first* n_{v_H} *columns of* P*. Also, assume that at most one punctured column has a non-zero integer entry in every row of* IRC *part. Furthermore, assume that the upper bound in* (1.5) *for* P *is a positive integer. Then, the same upper bound can be obtained with at most* $\binom{n_{v_H}}{n_{c_H}+1} \cdot (n_c + 1)$ *permanents, each of size at most* $(n_{c_H} + 1) \times (n_{c_H} + 1)$*.*

Proof. The proof is presented in Appendix 1.9.1. □

Our replacement strategy throughout the proofs so far has been of the following manner: If S does not contain the index of the column in P_{IR} with its only non-zero element in row ℓ, we seek columns (either punctured or unpunctured) indexed in S that have a non-zero element in row ℓ for replacement. Example 3 has only two such columns, both being punctured and identical. The example leads to the conclusion that this replacement strategy is not sufficient to show (if true) the complexity reduction that is possible if the PBRL protomatrix has more than one punctured column without the stricter conditions imposed in Theorem 5 when compared to Theorem 4. In fact, we can confirm that by considering only those sets that contain the indices of all the columns in P_{IR} we obtained the same upper bound for the protomatrix in Example 3 as the one obtained by considering all possible sets of size $n_c + 1 = 6$.

Based on other computational results we have obtained for protomatrices with more than one punctured column, it is our conjecture that the complexity reduction achieved so far in Theorems 3 and 4 holds true for the case when the protomatrix has more than one punctured column without having to impose stricter conditions than Theorem 4.

Note that the first assumption of Theorem 5 that every entry in the first n_{c_H} rows of all the punctured columns is non-zero is problematic for a code obtained from the HRC part. This is because such a code will never be able to pass useful information from any check node to any variable node during any iteration of the decoding process. In other words, the log-likelihood ratio values from all the check nodes to all the variable nodes will always be equal to zero. Therefore, PBRL codes satisfying this assumption are useless at the highest rate, but can be used as a code that corrects errors at lower rates provided that there is at least one check node that is connected to at most one punctured variable node.

1.4.4 Further Reduction in Complexity of the Design Procedure

In Sections 1.4.1, 1.4.2, and 1.4.3 we showed how the upper bounds of (1.4) and (1.5) for a PBRL protomatrix can be computed efficiently by only considering sets S that always contain the indices of all the columns of P_{IR}. These results yield a reduction in complexity of the design procedure while adding a new row to a PBRL protomatrix by reducing the amount and the complexity of computations required to obtain the upper bound for any candidate row under consideration. It turns out that the complexity of the overall design procedure can be further reduced the following way: Computations used in the design of a particular row of the IRC part can be reused while computing the upper bound for candidates considered in the design of the next row.

Consider a PBRL protomatrix P of size $n_c \times n_v$. Let $\mathcal{P} \triangleq \emptyset$ if P has no punctured columns. Let $n'_c \triangleq n_c - 1$ and $n'_v \triangleq n_v - 1$. If $n_c > n_{c_H}$, let P' denote the sub-matrix of the first n'_c rows and n'_v columns of P. P' is a PBRL protomatrix by itself. For any set $S \subseteq [n_v]$ with $|S| = n_c + 1$ let

$$\mathsf{sum}_{P,S} \triangleq \sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right). \tag{1.16}$$

Lemma 2. *Let P be a PBRL protomatrix of size $n_c > n_{c_H}$ and $n_v > n_{v_H}$. Let $S \subseteq [n_v]$, $|S| = n_c + 1$, and S contain the indices of all the columns in P_{IR} of P. Let $S' = S \setminus n_v$. Then,*

$$\mathsf{sum}_{P,S} = \mathsf{sum}_{P',S'} + \mathsf{perm}\left(P_{S \setminus n_v}\right). \tag{1.17}$$

Proof. As $P_S\left(i_1, n_c + 1\right) = 0$ for all $i_1 \neq n_c$ and $P_S\left(i_1, n_c + 1\right) = 1$ when $i_1 = n_c$ and $P_S\left(i_1, i_2\right) = P'_{S'}\left(i_1, i_2\right)$, $\forall i_1 \in \left[n'_c\right], i_2 \in [n_c]$, expanding the permanent of $P_{S \setminus i}$ when $i \neq$

$n_v, i \in S \setminus \mathcal{P}$ along n_c^{th} column of the sub-matrix $P_{S\setminus i}$ yields

$$\mathsf{perm}\left(P_{S\setminus i}\right) = 1 \cdot \mathsf{perm}\left(P'_{S'\setminus i}\right).$$

□

Furthermore when S contains the indices of all the columns of P_{IR}, $\mathsf{perm}\left(P_{S\setminus n_v}\right)$ depends only on elements $P(i,j)$ such that $i \in [n_{c_H}]$ or $i = n_c$ and $j \in S \setminus \{n_{v_H}+1, n_{v_H}+2, \ldots, n_v\}$.

Lemma 3. *Let P be a PBRL protomatrix with $n_c > n_{c_H}$ and $n_v > n_{v_H}$. Assume that P satisfies the assumptions of Theorem 3 if $n_p = 0$, satisfies the assumptions of Theorem 4 if $n_p = 1$, and satisfies the assumptions of Theorem 5 if $n_p > 1$. Let $\mathsf{sum}_{P,S}$ be as defined in Lemma 2. If we know the values of $\mathsf{sum}_{P',S'}$ for every $S' \subseteq [n'_v]$ such that $|S'| = n_c$ and $\{n_{v_H}+1, n_{v_H}+2, \ldots, n'_v\} \subseteq S'$, then the upper bound for P can be computed with exactly $\binom{n_{v_H}}{n_{c_H}+1}$ additional permanents beyond what has already been computed for P', each of size $(n_{c_H}+1) \times (n_{c_H}+1)$.*

Proof. Due to Theorems 3, 4, and 5 and Lemma 2. □

Remark 1. *The result of Lemma 3 implies the following for the design procedure: We can reuse the values of $\mathsf{sum}_{P',S'}$ for all S' such that $|S'| = n_c$ for which $\{n_{v_H}+1, n_{v_H}+2, \ldots, n'_v\} \subseteq S'$ and compute only one additional permanent from columns in P for each set $S = S' \cup \{n_v\}$ in order to obtain the upper bound for protomatrix P. The value of this permanent $\mathsf{perm}\left(P_{S\setminus n_v}\right)$ depends upon only the entries in the first n_{c_H} rows and n_c^{th} row of P and the $n_{c_H}+1$ columns indexed in S from the first n_{v_H} columns of P. Denote by γ a candidate vector of length n_{v_H} to be considered in the design of any row of the IRC part, P_{IRC}. Denote by P_t the matrix formed by appending γ as row $n_{c_H}+1$ to P_{HRC}. The result of Lemma 3 means that we need to compute $\binom{n_{v_H}}{n_{c_H}+1}$ permanents out of sub-matrices of P_t for such a candidate γ and the results of these computations can be used during the design of any row of P_{IRC} if γ is a candidate for that row. Therefore, the number of permanents to be computed* *for the design of P_{IRC} depends upon the number of candidates considered in the overall design procedure and not on the number of rows to be designed. The complexity of computing each permanent for any such candidate is exactly the complexity of computing the permanent of a matrix of size $(n_{c_H}+1) \times (n_{c_H}+1)$.*

Summary of algorithm to implement the simplified design procedure: We now summarize the overall, simplified design procedure to obtain a PBRL protomatrix of size $n_c \times n_v$ in Algorithm 1. Algorithm 1 presents the overall procedure for finding a PBRL protomatrix that maximizes (1.4) or (1.5) for each rate. It can be used to design a PBRL protomatrix whenever the conditions of either Theorem 3, 4, or 5 are satisfied in the PBRL design. The algorithm is presented in a way that the input is the HRC part and the output is the obtained PBRL protomatrix P. It can be modified and re-written as a recursive program that takes as its input a PBRL protomatrix and returns only one new row. The algorithm computes $\binom{n_{v_H}}{n_{c_H}+1}$ permanents of size $(n_{c_H}+1) \times (n_{c_H}+1)$ exactly once for every candidate under consideration. This is done in lines 11 through 17 of Algorithm 1. Furthermore the algorithm computes at most $\binom{n_{v_H}}{n_{c_H}+1}(n_{c_H}+1)$ permanents of size $n_{c_H} \times n_{c_H}$ that are candidate independent and are permanents required to compute the upper bound for the initial HRC matrix P_{HRC}. This is done in lines 19 through 27 of the algorithm that is presented. Lines 29 through 48 obtain the optimal candidate row at each design rate by using simple vector additions and by reusing the permanent computations of lines 11 through 17 and lines 19 through 27.

The following are some of the variable identifiers introduced in Algorithm 1: Assume that the candidates for each row of P_{IRC} are sets of vectors of length n_{v_H}. Let the set of candidates for row $i \in [n_c - n_{c_H}]$ of P_{IRC} be denoted C_i and let $C = \cup_i C_i$. Following Remark 1, let SumMatrix be a matrix that stores the required $\binom{n_{v_H}}{n_{c_H}+1}$ permanents, row-wise, for each candidate (there are, in total, $|C|$ candidates). Let PunctPattern be a vector of length n_{v_H} that specifies whether a variable node among the first n_{v_H} is punctured or transmitted, with 0 indicating that the variable node is punctured. Other variable identifiers introduced

Algorithm 1 Permanent-bound-based PBRL design

```
Function perm in the following, which computes the permanent of a square matrix, can be implemented using Ryser's algorithm [Rys63]
Inputs, Output: The main function Pbdesign of the algorithm takes as inputs $P_{\mathsf{HRC}}$, $n_c$, and $\{C_i, i \in [n_c - n_{c_H}]\}$ and returns protomatrix $P$ as the output
function MatrixOut = Enumerate$(a, b)$
    Return a matrix of size $\binom{a}{b} \times b$ containing all $b$ combinations (without repetition) of $[a]$
end function
function $P$ = Pbdesign($P_{\mathsf{HRC}}$, $n_c$, $\{C_i, i \in [n_c - n_{c_H}]\}$, PunctPattern)
    $C \leftarrow \cup_i C_i$
    Assign an index $1 \le i \le |C|$ to each $\gamma$ in $C$
    Combinations $\leftarrow$ Enumerate $(n_{v_H}, n_{c_H} + 1)$
    SumMatrix $\leftarrow$ ZeroMatrix $\left(|C|, \binom{n_{v_H}}{n_{c_H}+1}\right)$
    for $i \leftarrow 1 : |C|$ do
        $P_t \leftarrow$ append $\gamma_i$ as new row to $P_{\mathsf{HRC}}$
        for $j \leftarrow 1 : \binom{n_{v_H}}{n_{c_H}+1}$ do
            $S \leftarrow$ row $j$ of Combinations
            SumMatrix$(i, j) \leftarrow$ perm $(P_{t_S})$
        end for
    end for
    OptimalVector $\leftarrow$ ZeroMatrix $\left(1, \binom{n_{v_H}}{n_{c_H}+1}\right)$
    for $i \leftarrow 1 : \binom{n_{v_H}}{n_{c_H}+1}$ do
        $S \leftarrow$ row $i$ of Combinations
        for $j \leftarrow 1 : n_{c_H} + 1$ do
            if PunctPattern$(S(j)) == 0$ then
                Continue
            end if
            OptimalVector$(i) \leftarrow$ OptimalVector$(i)$ + perm $\left(P_{\mathsf{HRC}_{S \setminus S(j)}}\right)$
        end for
    end for
    $P \leftarrow P_{\mathsf{HRC}}$
    for $i \leftarrow 1 : n_c - n_{c_H}$ do
        UBOptimal $\leftarrow 0$
        for $j \leftarrow 1 : |C|$ do
            if $\gamma_j \notin C_i$ then
                Continue
            end if
            TempVector $\leftarrow$ OptimalVector + row $j$ of SumMatrix
            if UBOptimal $== 0$ then
                UBOptimal $\leftarrow$ min(TempVector)
                OptimalCandidate ← $\gamma_j$
                OptimalCandidateSums ← TempVector
            **else if** UBOptimal < min(TempVector) **then**
                UBOptimal ← min(TempVector)
                OptimalCandidate ← $\gamma_j$
                OptimalCandidateSums ← TempVector
            **end if**
        **end for**
        OptimalVector ← OptimalCandidateSums
        $P$ ← append OptimalCandidate as new row to $P$
    **end for**
    $P$ ← append 0 and $I$ matrices to $P$
    Return $P$
**end function**
```

in the algorithm are self-explanatory.

We note that the computational complexity of Algorithm 1 is extremely low when compared to the threshold-based PBRL design approach of [CVDW15]. This is because it usually requires more than 500 iterations of message passing and real-valued function computations in order to compute the threshold of a given PBRL protomatrix using the RCA approach [CVDW15]. Moreover, one cannot reuse computations performed during a run of the RCA algorithm at a higher rate to find the optimal row for the next lower rate.

1.5 Upper Bound on the Optimal Value of the Exhaustive Search for a Row of IRC Part

In Section 1.4 we showed how the overall design procedure is simplified because of the presence of the identity matrix structure in the columns of P_{IR} in a PBRL protomatrix. In particular, we presented Algorithm 1 that performs the following for a given HRC matrix (assuming that the conditions of either of Theorems 3, 4, or 5 are satisfied during the design process): First, compute an $\binom{n_{v_H}}{n_{c_H}+1}$-length vector of permanents of size $(n_{c_H}+1)\times(n_{c_H}+1)$ for each candidate. Then, during the design of any row add the vector of permanents corresponding to the candidate under consideration to a pre-computed vector of the same length obtained from the design of the previous row of the protomatrix. The minimum value across this new resultant vector is the upper bound on minimum distance for the protomatrix with the candidate under consideration as the new row.

In this section, we show that the problem of selecting an optimal candidate for a row of a PBRL protomatrix can be cast as an integer linear program (ILP) whenever the assumptions of either of Theorems 3, 4, or 5 are satisfied. Apart from being an interesting equivalent view of the design process, the ILPs let us obtain bounds on the optimal value of the minimum distance upper bound that can be obtained by choosing the best possible candidate from a feasible set of candidates for a new row. This equivalent formulation of the exhaustive search to find an optimal row also shows that the search, in general, is a computationally complex problem.

1.5.1 Linear Programming (LP) Upper Bound

For a PBRL protomatrix P of size $n_c \times n_v$, assume the definitions of n'_c, n'_v, $\mathsf{sum}_{P,S}$, and P' as previously stated in Section 1.4.4.

Let x be a vector of length n_{v_H} that denotes the new P_{IRC} row being designed for a PBRL protomatrix P. Following the results of Section 1.4.4, let b be a vector of length $\binom{n_{v_H}}{n_{c_H}+1}$ that contains the $\binom{n_{v_H}}{n_{c_H}+1}$ values of $\mathsf{sum}_{P',S'}$, one for each possible S'. Let A be a matrix of size $\binom{n_{v_H}}{n_{c_H}+1} \times n_{v_H}$ whose rows comprise the following: For the set $S_i = S'_i \cup \{n_v\}$ with S'_i corresponding to $b_i = \mathsf{sum}_{P',S'_i}$, let $\bar{S}_i = S_i \setminus \{n_{v_H}+1, n_{v_H}+2, \ldots, n_v\}$. That is, $\bar{S}_i$ contains the $n_{c_H}+1$ indices in S_i that are chosen, without repetition, from indices $\{1, 2, \ldots, n_{v_H}\}$. Now, entry $A(i,j)$ is defined as

$$A(i,j) \triangleq \begin{cases} \mathsf{perm}\left(P_{\mathsf{HRC}_{\bar{S}_i \setminus j}}\right), & j \in \bar{S}_i, \\ 0, & j \notin \bar{S}_i, \end{cases} \tag{1.18}$$

where the permanents computed here are of size $n_{c_H} \times n_{c_H}$.

With a candidate vector x as the new row, the minimum distance upper bound, assuming that either of Theorems 3, 4, or 5 holds for the design process, can thus be computed as

$$\min \;\; Ax + b, \tag{1.19}$$

where the minimum is over the elements of the vector $Ax + b$. The exhaustive search over all feasible vectors x can now be represented as

$$\begin{aligned} \max_{x} \quad & \min Ax + b, \\ \text{s.t.} \quad & x \text{ a feasible vector}. \end{aligned} \tag{1.20}$$

Upon choosing an optimal vector x^* from the feasible set, the new b vector for the design of the next row is obtained as $b \leftarrow Ax^* + b$; matrix A remains unchanged.

The search can now be equivalently re-written as

$$\begin{aligned} -\min \quad & \beta, \\ \text{s.t.} \quad & -(Ax + b) \leq \beta \cdot \mathbf{1}, \\ & x \text{ a feasible vector}, \end{aligned} \tag{1.21}$$

where β is a real scalar and $\mathbf{1}$ is a vector of all ones of appropriate size.

The constraints that define the feasible set of x determine whether exhaustive search (1.21) is a convex optimization problem or not (See [BV09] for convex optimization). In order for the search process (1.21) to be exact, we require that the design process satisfy the conditions of either of Theorems 3, 4, or 5 according to the presence of $n_p = 0$, $n_p = 1$, or $n_p > 1$ punctured variable nodes, respectively. Theorems 3 and 4 impose constraints only on the HRC part of the PBRL protomatrix, whereas Theorem 5 contains assumptions about the candidates that can be chosen for each row of the IRC part. For simplicity, we deal only with the former two cases in the following.

Theorems 3 and 4 do not implicitly assume constraints about x, but we can impose reasonable constraints that suit the design for PBRL families that support short blocklengths. In particular, as we will further see in design examples of Section 1.6, we can impose constraints such as the following:

$$w_{\min} \leq \sum_{i=1}^{n_{v_H}} x_i \leq \ w_{\max}, \tag{1.22}$$

$$w_{i,\min} \leq x_i \leq \ w_{i,\max}, \ \ 1 \leq i \leq n_{v_H}, \tag{1.23}$$

$$x_i \in \ \mathbb{Z}, \ 1 \leq i \leq n_{v_H}, \tag{1.24}$$

where the constraint that x_i take on integer values is necessary for the design process and prevents (1.21) from being a convex optimization problem by forcing the feasible set to be non-convex. The above constraints are only a representative of the types of linear inequality constraints that can be imposed. These constraints turn search process (1.21) into the following integer linear program (ILP):

$$\begin{aligned} -\min \ \ & \beta, \\ \text{s.t.} \ \ & -(Ax+b) \leq \beta \cdot \mathbf{1}, \\ & Cx \leq d, \\ & x_i \in \mathbb{Z}, \ \ 1 \leq i \leq n_{v_H}, \end{aligned} \tag{1.25}$$

where C is a matrix and d is a vector that we have introduced to represent the linear constraints on x. Relaxing the integer constraint on values of x_i now would lead to a (convex) linear program whose optimal value, which can be computed efficiently, provides an upper bound on the optimal value of the optimization problem in (1.25). In Section 1.6, we present numerical examples of computation of the optimal values of the LP relaxation alongside numerical results obtained through the design process of Algorithm 1.

Remark 2. *A significance of the optimal values of the LP relaxation derived in this section* *is the following. The LP upper bounds are a by-product of the existing permanent-based upper bounds that this chapter is based on. The bounds obtained by solving the LP relaxation represent a new set of minimum distance upper bounds that one can obtain for the specific class of PBRL protomatrices (under the reduced complexity scenarios) without even having to construct a PBRL protomatrix.*

1.6 Numerical Design Examples

In this section we design PBRL protomatrices according to the new design method we have proposed. We assume the following HRC matrix for designs in this section:

$$P_{\mathsf{HRC},1} = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{bmatrix} \tag{1.26}$$

We consider both a punctured and an unpunctured version of $P_{\mathsf{HRC},1}$ in our examples. The punctured version has the first variable node punctured. Hence the design rate we start with is either 6/8 (unpunctured) or 6/7 (punctured).

The HRC matrix in (1.26) has a minimum distance upper bound of 12 when none of its variable nodes are punctured and an upper bound of 8 when its first column is punctured. One can verify via Lemma 1, therefore, that any unpunctured or punctured PBRL protomatrix that can be obtained from $P_{\mathsf{HRC},1}$ will satisfy the conditions of Theorems 3 or 4, respectively. Hence, Lemma 3 applies in this case and we can use Algorithm 1 to design PBRL protomatrices using $P_{\mathsf{HRC},1}$ when at most one of its columns is punctured.

Remark 3 (Design constraints)**.** *We constrain the last $n_c - n_{c_H}$ rows of the protomatrix to have a weight of exactly 4 and do not allow any non-zero integer other than 1. These constraints facilitate good performance at short block-lengths because limiting both the density and the number of multiple edges in the protograph helps the resulting LDPC codes have good girth upon lifting and avoid having too many short cycles. Explicit constraints are necessary* *because increasing the value of any element in any position of a protomatrix with a finite permanent bound of* (1.4) *or* (1.5) *either results in an increase in the bound or the bound stays the same. In other words, the problem of selecting an optimal row is trivial without such constraints.*

Remark 4. *Our design method, which maximizes an upper bound on code minimum distance, does not depend upon the channel over which we use the codes. For designing codes for comparison according to the original PBRL design method that involves computing iterative decoding thresholds, we assume the binary-input additive white Gaussian noise channel (BI-AWGNC). The threshold values computed in this chapter are the result of at least 1000 iterations of the reciprocal channel approximation (RCA) method (See [CVDW15]).*

Unpunctured design

We design two unpunctured ensembles using the HRC matrix given in (1.26) with the design constraints in Remark 3. The design rates we consider decrease from 6/8 to 6/15. The first ensemble, P_1, is obtained using the new permanent bound design (PBD) method proposed in this chapter. For comparison, the second ensemble, P_2, is designed by optimizing the iterative decoding thresholds (referred to as "threshold-based" in results) over BI-AWGNC. The IRC parts of P_1 and P_2 that we obtained are:

$$P_{1,\mathsf{IRC}} = \begin{bmatrix} 1\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0 \\ 1\,0\,0\,0\,0\,0\,1\,1 \\ 0\,1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,0\,1\,1\,0\,0 \\ 1\,0\,1\,0\,0\,0\,0\,1 \\ 0\,1\,0\,1\,0\,0\,1\,0 \end{bmatrix}, P_{2,\mathsf{IRC}} = \begin{bmatrix} 1\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0 \\ 1\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,1\,0\,1\,0\,1\,0 \\ 1\,0\,0\,1\,1\,0\,0\,0 \\ 1\,0\,1\,0\,0\,1\,0\,0 \\ 1\,0\,1\,0\,0\,0\,1\,0 \end{bmatrix} \qquad (1.27)$$

Punctured design

Similarly, we design two punctured ensembles using the HRC matrix in (1.26) via the design constraints in Remark 3. The first variable node is punctured[2], and the design rates decrease from 6/7 to 6/15. The resulting ensembles are called P_3 (PBD) and P_4 (threshold-based):

$$P_{3,\mathsf{IRC}} = \begin{bmatrix} 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0 \\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1 \\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0 \\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1 \\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0 \end{bmatrix}, P_{4,\mathsf{IRC}} = \begin{bmatrix} 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0 \\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0 \\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0 \\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0 \\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0 \end{bmatrix} \qquad (1.28)$$

Unconstrained design via original PBRL design method

For further comparison, we design an ensemble, called P_5, using the same HRC matrix with its first variable node punctured according to the original PBRL design method. The design rates decrease from 6/7 to 6/15. For this ensemble we only have the following restriction in the IRC part: We do not permit any integer greater than 1. The design yielded the following

[2] A punctured variable node improves the iterative decoding threshold (See [DDJA09] and [CVDW15]). Also, following the observations of [CVDW15] we constrain every row of P_{IRC} to connect to the punctured variable node in all of our punctured designs in this section.

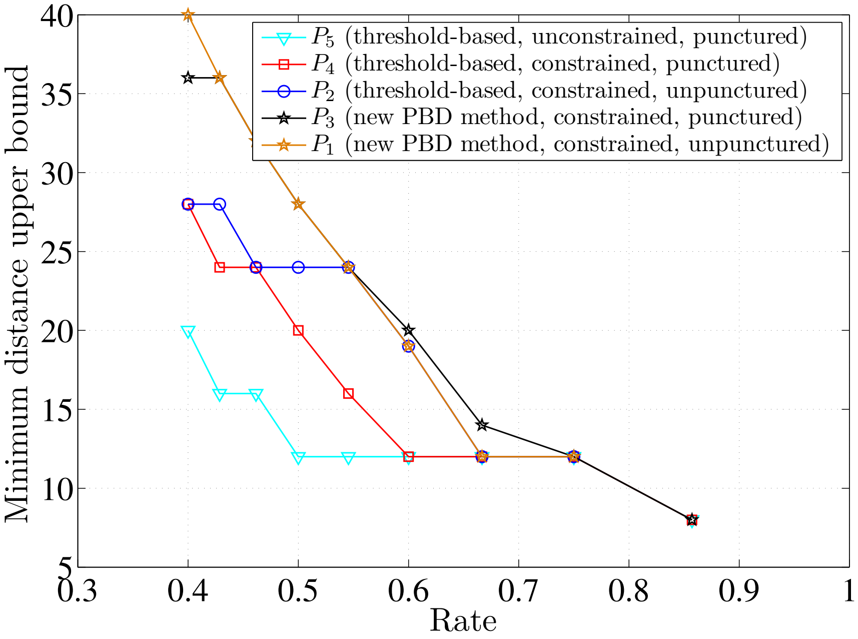

Figure 1.2: Comparison of minimum distance upper bounds of ensembles $P_i, i \in [5]$, obtained from HRC matrix $P_{\mathsf{HRC},1}$ in (1.26)

IRC part (note that some rows of P_{IRC} have a weight larger than 3):

$$P_{5,\mathsf{IRC}} = \begin{bmatrix} 1\,0\,1\,0\,1\,0\,0\,0 \\ 1\,0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0 \\ 1\,0\,1\,1\,1\,0\,0\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0 \\ 1\,0\,1\,1\,0\,0\,0\,0 \end{bmatrix} \tag{1.29}$$

We now compare the five ensembles $P_i, i \in [5]$, according to two design metrics. Fig. 1.2 shows the upper bound on the minimum distance obtained at each design rate for the five ensembles. P_1, the unpunctured, constrained ensemble obtained via the new PBD method, has the best upper bound at almost every design rate. At the other end of the spectrum, P_5, the unconstrained, punctured ensemble designed to optimize the threshold at each rate,

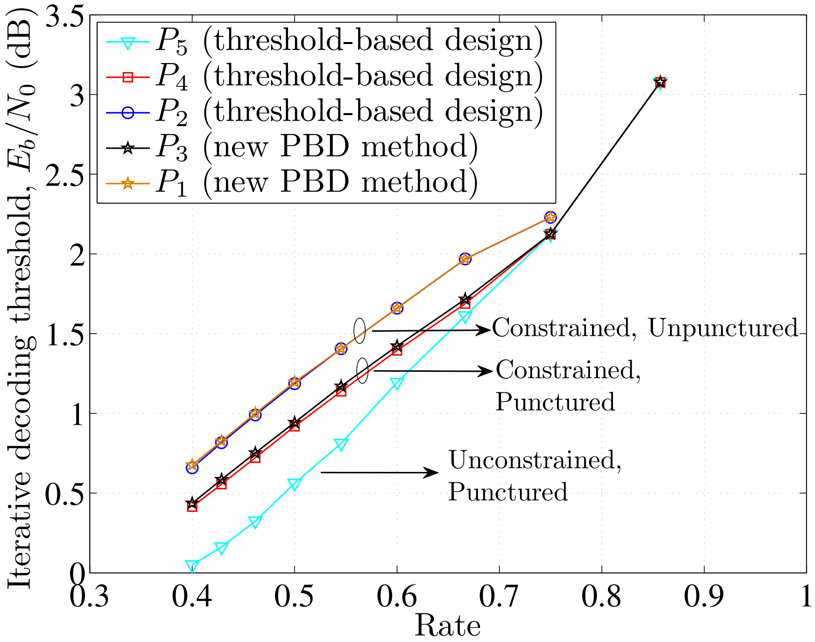

Figure 1.3: Iterative decoding thresholds over BI-AWGNC for ensembles $P_i, i \in [5]$, obtained from HRC matrix $P_{\mathsf{HRC},1}$

has the smallest upper bounds.

The iterative decoding thresholds at each design rate (over BI-AWGNC) for all five ensembles are shown in Fig. 1.3. As expected, ensemble P_5 has the best threshold at each design rate. But surprisingly, both the unpunctured and punctured constrained ensembles obtained via the new PBD method, P_1 and P_3, have almost the same threshold at each design rate as their counterpart ensembles P_2 and P_4, which were obtained by optimizing the iterative decoding threshold at each design rate. While this appears to be the case for the examples in this section, the examples of Section 1.7.3 show that such a property is not obtained always.

1.6.1 Comparison of LP Upper Bound and Optimal Values of Minimum Distance Upper Bound at Each Design Rate

We provide here an example of evaluating the relaxed version of the ILP in (1.25). This leads to an important conclusion regarding the optimality of the relaxed problem. For the punctured design $P_{3,\mathsf{IRC}}$ with HRC part (1.26) and its first column punctured, the matrix A is of size 56×8 and the vector b is of size 56×1. The design constraints introduced in Remark 3 and the requirement that the punctured variable node have a connection to every row of the IRC part lead to the following linear inequality constraints for the ILP in (1.25) to design every row:

$$\sum_{i=1}^{8} x_i \leq 3, \qquad 0 \leq x_i \leq 1, 1 \leq i \leq 8, \qquad x_1 = 1 \tag{1.30}$$

We solve the relaxed version of the ILP by removing the integer constraint $x_i \in \mathbb{Z}$ of (1.25). For every design rate we obtain two upper bounds by solving the LP relaxation, with the two upper bounds differing in how vector b is updated. The first set of upper bounds, called UB_1, at rates 6/8 through 6/15 is obtained by continuously updating b as $Ax^* + b$, where x^* is the optimal vector as obtained by solving the LP relaxation at each step. Note that the x^* that is provided by the relaxation can have non-integer values at some positions. For example, the vector x^* for the LP relaxation at design rate 6/8 is

$$[1, 0.2353, 0.3529, 0.2353, 0.3529, 0.2353, 0.3529, 0.2353].$$

The second set of upper bounds, called UB_2, is obtained by updating b with the x^* provided by the actual design process. That is, we compute the upper bound for the current design rate using the LP relaxation and update b for the next design rate using as x^* the optimal vector obtained in $P_{3,\mathsf{IRC}}$ at the current design rate.

The results of the two upper bounds on the optimal value of the permanent bound obtained using the LP relaxations and the actual optimal value at each design rate obtained through the design of $P_{3,\mathsf{IRC}}$ are shown in Fig. 1.4. A significant result of our experiment is that we observed that the x^* that we obtained while solving the LP relaxation contained non-integer values for most design rates and while computing both the upper bounds. This shows that the ILP, in general, does not possess a problem structure that would lead to

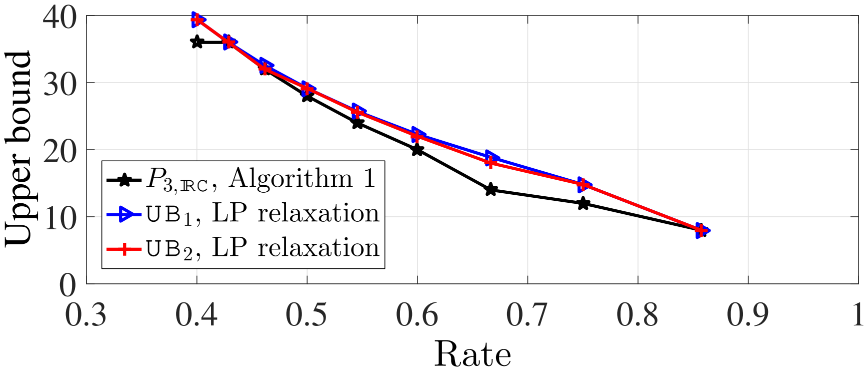

Figure 1.4: Comparison of optimal values of the upper bound from the design procedure that yielded $P_{3,\mathsf{IRC}}$ and from the LP relaxations of (1.25)

the conclusion that the LP relaxation is an exact formulation of the ILP. Nonetheless, the complexity reduction identified in this chapter allows Algorithm 1 to operate efficiently for most PBRL designs even though the algorithm is an exhaustive search.

1.7 Simulation Results and Discussion

In this section, Section 1.7.1 presents simulation results of carefully designed RC code families from the five protograph families $P_i, i \in [5]$, obtained from $P_{\mathsf{HRC},1}$ in Section 1.6. Further, we also present similar results in Section 1.7.3 for three ensembles that we designed using another HRC matrix that we call $P_{\mathsf{HRC},2}$:

$$P_{\mathsf{HRC},2} = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & 1 & 2 & 1 \\ 1 & 3 & 1 & 3 & 1 & 2 & 1 & 2 \end{bmatrix} \tag{1.31}$$

$P_{\mathsf{HRC},2}$ is irregular in its degree distribution, whereas $P_{\mathsf{HRC},1}$ (in (1.26)) is regular.

Remark 5 (Lifting and simulation parameters)**.** *Codes simulated in this section are all quasi-cyclic. Lifting was performed for the protomatrix of the lowest rate* $6/15$ *using the circulant-PEG (C-PEG) algorithm combined with the ACE algorithm of Tian et al. [TJVW04]. The ACE algorithm guarantees that all cycles of length up to* $2d_{ACE}$ *have an approximate cycle extrinsic degree (ACE) value of at least* η_{ACE} *(See [TJVW04] for ACE). The lifting algorithm* *that produced the codes in this chapter was constrained to satisfy a target girth of at least 6 and a target* $(2d_{ACE}, \eta_{ACE})$ *pair value that is as high as possible for a given protomatrix. The lifting factor used is 33, which resulted in* $k = 198$ *information bits for all code families. Simulation results shown were obtained using a maximum of 100 iterations of full-precision, flooding, LLR-domain belief propagation [RL09] over BI-AWGNC. At least 100 errors were collected for each frame error rate (FER) point in any simulated* E_b/N_0 *vs. FER graph.*

1.7.1 Frame Error Rates of QC Code Families from $P_i, i \in [5]$

First, we note that the $(2d_{ACE}, \eta_{ACE})$ values that we obtained during lifting of the lowest-rate protographs $P_i, i \in [5]$, are as follows: $P_1 : (16, 10)$, $P_2 : (16, 8)$, $P_3 : (16, 9)$, $P_4 : (16, 8)$, $P_5 : (16, 4)$. The target cycle length $2d_{ACE}$ was set to 16 in all cases, which is sufficiently large that the impact of the ACE values on low-FER performance is dominated by the guaranteed extrinsic message degree η_{ACE}. The best ACE values were achieved by the new PBD designs P_1 with $\eta_{ACE} = 10$ and P_3 with $\eta_{ACE} = 9$. The constrained, threshold-based PBRL designs had the next best ACE values with $\eta_{ACE} = 8$, and the smallest ACE value was achieved by the unconstrained, threshold-based design of P_5 with $\eta_{ACE} = 4$.

The FER performance of the QC PBRL code families obtained from these ensembles are shown in Figs. 1.5 and 1.6. Note that the low-FER performance for the lowest rate (rate 6/15) codes is consistent with the ACE ordering described in the previous paragraph. The QC-LDPC code family obtained from constrained, punctured, PBD ensemble P_3 outperforms all the other code families at FERs 10^{-4}, 10^{-5}, and 10^{-6} and at all rates (including at rates 6/11, 6/12, and 6/13, which are omitted from the figures to avoid clutter). This ensemble has the advantages of a good, if not the best, threshold due to the punctured variable node (See Fig. 1.3.) and a good upper bound on the minimum distance at all rates (See Fig. 1.2). P_1, which has the best ACE value at rate 6/15, performs better than P_3 at even lower FERs at rate 6/15.

The QC PBRL code family from the constrained, unpunctured, PBD ensemble P_1, which

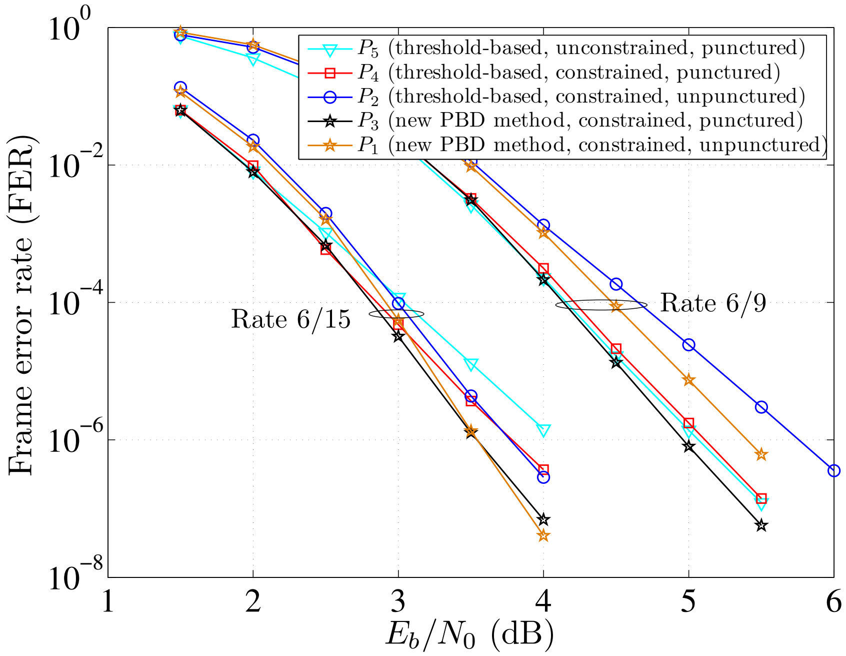

Figure 1.5: Comparison of FER performance of codes obtained from ensembles $P_i, i \in [5]$, at rates 6/15 and 6/9

has the best minimum distance upper bound at all but two rates, performs well at lower rates but not at higher rates. For example, at a higher rate of 6/10, the code from P_1 has an FER performance curve that is above the codes from P_3, P_4, and P_5. At this rate, the minimum distance upper bound of P_1 becomes closer to that of the other ensembles (See Fig. 1.2.) and performance is affected by the fact that P_1 possesses a threshold that is not competitive enough (See Fig. 1.3.) to deliver a waterfall curve that is competitive against the punctured ensembles. However, the slope of the FER curve of the rate-6/10 code from P_1 at low FERs, as shown in Fig. 1.6, is suggestive of a performance that is better than that of the threshold-based ensembles P_2, P_4, and P_5 at lower FERs.

The ensembles with the three best thresholds across all rates are P_3, P_4, and P_5, which are all punctured. Of the three, the code family obtained from P_3 resulted from the new design method. It can be seen from the results in Figs. 1.5 and 1.6 that, even though P_5 has the best threshold across all design rates (See Fig. 1.3.) the codes from the P_5 ensemble have a frame error rate slope that is not competitive when compared to P_3. Code family

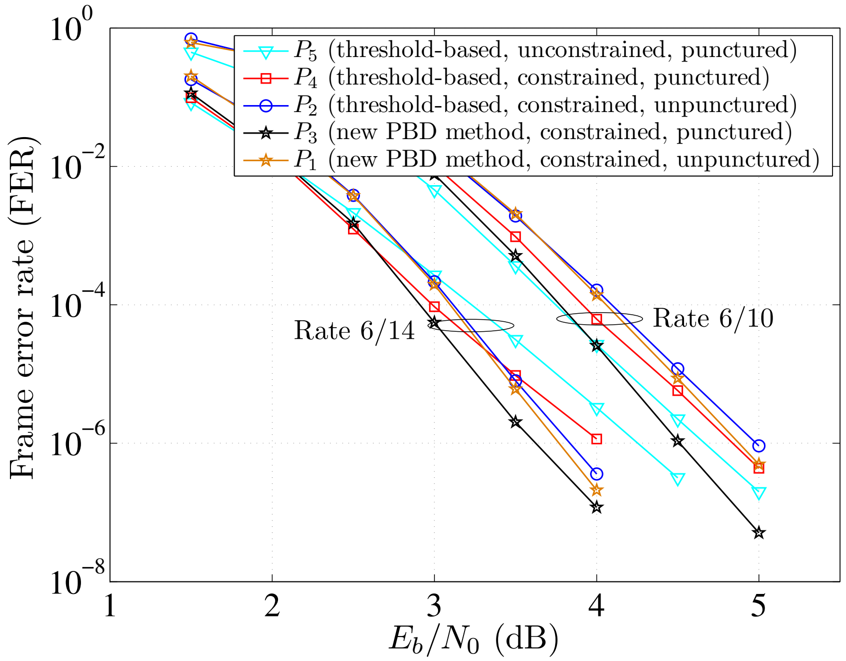

Figure 1.6: Comparison of FER performance of codes obtained from ensembles $P_i, i \in [5]$, at rates 6/14 and 6/10

P_4 is also not competitive with respect to P_3 at low FERs despite both ensembles having virtually the same set of thresholds across all design rates. Here we see the benefit of having a better permanent bound when thresholds are similar.

The gap to BI-AWGNC capacity at an FER of 10^{-6} is shown in Fig. 1.7. The QC PBRL code family of ensemble P_3 achieves the best performance at all design rates. The performance of this code family is about 1.5 dB away at all rates from the refined normal approximation of [PPV10]. The figure shows that this code family is the only one of the five to have a constant gap to BI-AWGNC capacity at all the design rates.

Recall from Fig. 1.3 that the newly designed PBRL protomatrices P_1 and P_3 have more or less the same threshold as the threshold-based PBRL protomatrices P_2 and P_4, respectively. Thus, both designs could have been equally chosen if threshold were the design metric. However, looking at Fig. 1.2, we see that for lower rates P_1 has a better permanent bound than P_2, and that P_3 has a better permanent bound than P_4. Looking at Fig. 1.7 we see that the PBD method did a better job of identifying rate-compatible ensembles that produce QC

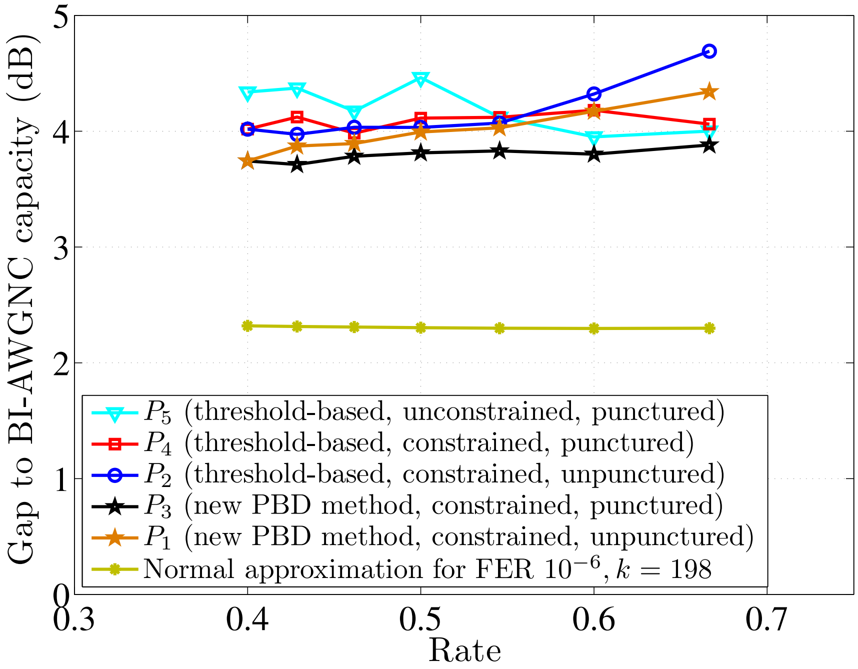

Figure 1.7: Comparison of gap to BI-AWGNC capacity at a frame error rate (FER) of 10^{-6} and information block size of $k = 198$ for QC codes from the ensembles $P_i, i \in [5]$, and the refined normal approximation of [PPV10]

code families with improved low-FER performance at short block-lengths.

1.7.2 Empirical Codeword Weight and Non-Convergence Spectra

In order to better understand the performance gains we obtain through the PBD method, we analyzed the empirical codeword weight spectrum and the non-convergence set spectrum of the codes we constructed. An *undetectable error* in the iterative decoding process occurs when the decoder converges to an incorrect codeword. A *detectable error* occurs when the decoder does not converge to any codeword within the given maximum number of iterations. For every simulation instance, iterative decoding terminates with one of three outcomes: the decoder converges to the correct codeword (no error), or the decoder does not converge (detectable error), or the decoder converges to an incorrect codeword (undetectable error).

In case the decoder converges to an undetectable error, we make note of the weight of the incorrect codeword to which the decoder converged. The empirical frequencies with

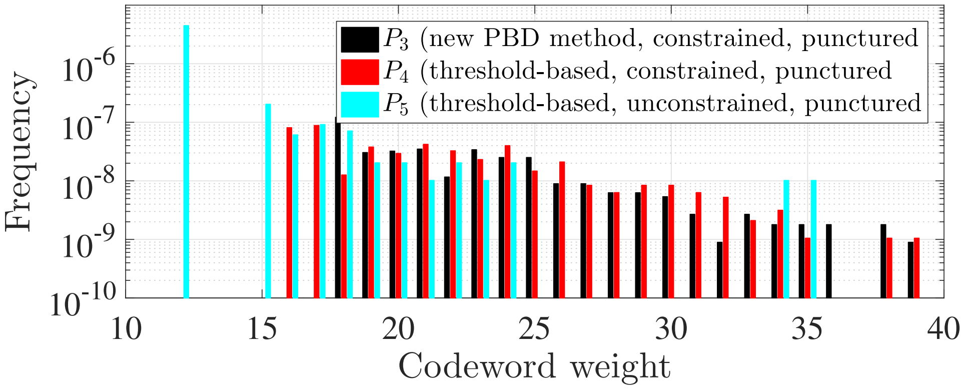

Figure 1.8: Comparison of frequency of convergence to incorrect codewords of the rate-6/12 code from ensembles P_3, P_4, and P_5 at an E_b/N_0 of 3.5dB

which the decoder incorrectly converges to the possible (codeword) weights is the empirical codeword weight spectrum.

A detectable error usually occurs due to non-codewords in an LDPC code such as stopping sets, trapping sets, etc. It is a difficult problem to enumerate or identify all non-codewords, but we capture the essence of these graphical objects through what we call the empirical non-convergence set spectrum. For each simulation instance, we track the weight of the smallest set of variable nodes that fail to converge to their correct values and report this weight for simulations that culminated in non-convergence. We call the smallest such non-empty set in every simulation instance the non-convergence set. In case a detectable error occurs, it means that the decoder has operated over its maximum number of iterations and at the end of every iteration the decoder encountered some variable nodes that are in error. The empirical frequencies with which the decoder "gets stuck" in non-convergence sets of various weights is the empirical non-convergence set spectrum.

Figs. 1.8 and 1.9 depict the empirical codeword weight spectrum and non-convergence set spectrum for the rate-6/12 code obtained from the punctured ensembles P_3, P_4, and P_5 at an E_b/N_0 of 3.5dB. It is seen from these figures that the code from ensemble P_3 possesses the most favorable spectra of the three in terms of the graphical structures that dictate low-FER performance. P_3 was obtained using the PBD method, which maximizes an upper bound on the minimum distance.

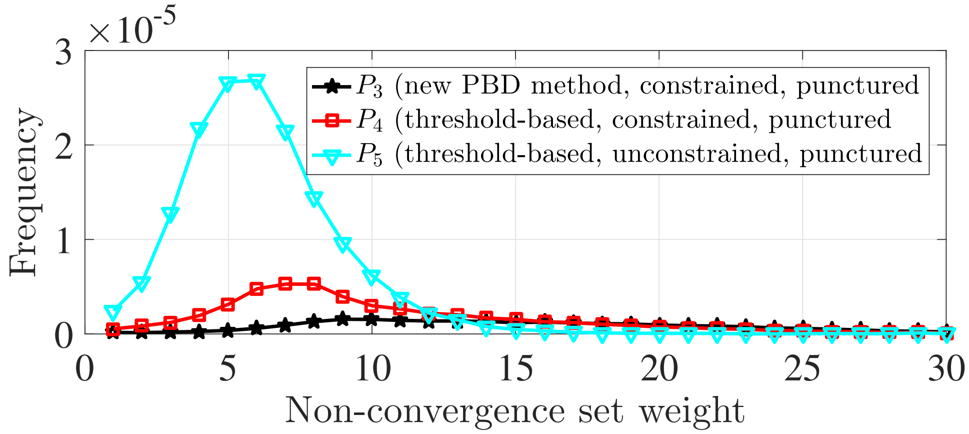

Figure 1.9: Comparison of frequency of occurrence of non-convergence sets that lead to non-convergence for the rate-6/12 code from ensembles P_3, P_4, and P_5 at an E_b/N_0 of 3.5dB

Table 1.1: Minimum Distance Upper Bounds (in parenthesis) and Empirical Smallest Codeword Weights Obtained from Simulations of Code Families from $P_i, i \in [5]$, at Five Design Rates

Ensemble	Design rate				
	6/10	6/11	6/12	6/14	6/15
P_1	11 (19)	13 (24)	17 (28)	21 (36)	24 (40)
P_2	11 (19)	14 (24)	16 (24)	21 (28)	22 (28)
P_3	13 (20)	15 (24)	18 (28)	20 (36)	21 (36)
P_4	12 (12)	14 (16)	16 (20)	18 (24)	21 (28)
P_5	12 (12)	12 (12)	12 (12)	16 (16)	20 (20)

Note that the permanent bound is an upper bound on the distance, and the actual minimum distance of the code will often be below this bound. Table 1.1 shows the minimum distance upper bounds (in parenthesis) along with the empirical smallest codeword weights obtained from simulation for the five code families at five design rates.

Any upper bound on the minimum distance is also an upper bound on the size of the smallest stopping set of the code and more generally on the smallest size of other non-convergence sets. This is a critical relationship at short block-lengths because the more small stopping sets (and other such objects) there are in a code, the more often the decoder gets stuck in configurations that lead to detectable errors. The point of the new code design method for short block-lengths is not that the designed codes will have the permanent bound as their distance. The goal is merely that by seeking to maximize this bound we achieve a better codeword weight spectrum and better non-convergence set spectrum than would be obtained by focusing only on threshold as a design metric.

Remark 6. *Note that PBRL ensembles P_1 and P_2 have exactly the same permanent bound for rates $6/8$, $6/9$, $6/10$, and $6/11$ in Fig. 1.2. However, comparing the FER performance for the codes from these two ensembles at rates $6/9$ and $6/10$ in Figs. 1.5 and 1.6, respectively, we see that the codes from P_1 perform better than those from P_2. To understand this difference, recall that the permanent bound that we compute is based entirely on the protograph and that two codes with the same upper bound, and indeed even the same protograph, can have different actual minimum distances and more generally different codeword weight spectra and non-convergence set spectra.*

In the case of P_1 and P_2 in particular, we note that these two codes do share the same protograph for all rates greater than or equal to 6/11. However, P_1 and P_2 do not yield identical codes for these rates because they were lifted differently. Our code design approach only lifts the lowest-rate code, which implies the lifting of all higher rates. To investigate this further, we ran a simulation to determine the empirical minimum distance for these rate-6/10 codes and found that both codes had an empirical minimum distance of 11 but that the code from P_1 had a superior non-convergence set spectrum. In this case, the difference in performance between P_1 and P_2 cannot be attributed to the protograph, but rather to a better lifting. It is possible that a better lifting was facilitated by P_1's superior permanent bound at the lowest rate.

While the above remark shows that two codes can have the same permanent bound (and even the same protograph) but different performance after lifting, we emphasize that our experiments have shown that when two codes have different permanent bounds, the code with the superior permanent bound, provided its threshold is competitive, has displayed a better low-FER performance.

1.7.3 Simulation Results for Ensembles Obtained using $P_{\mathsf{HRC},2}$

We first provide a brief overview of the ensembles obtained using $P_{\mathsf{HRC},2}$ (in (1.31)). For any ensemble that has a property of "constrained", we follow the same constraint we imposed while designing ensembles using $P_{\mathsf{HRC},1}$ in Section 1.6. That is, we constrain the last $n_c - n_{c_H}$ rows of the protomatrix to have a weight of exactly 4 and do not allow any non-zero integer other than 1 in these rows. Similarly, an ensemble that is "punctured" has its first column punctured and P_{IRC} is connected to the punctured column in every row.

The following are the IRC parts of the punctured and constrained ensembles obtained using $P_{\mathsf{HRC},2}$, where P_6 is based on PBD method and P_7 is threshold-based:

$$P_{6,\mathsf{IRC}} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,0\,0\,1\,1 \\ 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1\,1\,0\,0 \\ 1\,0\,0\,1\,0\,0\,1\,0 \\ 1\,0\,0\,0\,0\,1\,1\,0 \\ 1\,0\,0\,0\,1\,0\,0\,1 \end{bmatrix}, \quad P_{7,\mathsf{IRC}} = \begin{bmatrix} 1\,1\,0\,1\,0\,0\,0\,0 \\ 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,0\,1\,0\,1\,0 \\ 1\,0\,0\,0\,1\,0\,1\,0 \\ 1\,0\,1\,0\,0\,1\,0\,0 \\ 1\,0\,1\,0\,0\,0\,0\,1 \\ 1\,1\,0\,1\,0\,0\,0\,0 \end{bmatrix} \tag{1.32}$$

The following is the IRC part of the threshold-based, punctured, and unconstrained ensemble

P_8:

$$P_{8,\mathsf{IRC}} = \begin{bmatrix} 1\,0\,1\,0\,0\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0\,0\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0\,1\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 1\,0\,1\,0\,0\,0\,1\,0 \end{bmatrix} \tag{1.33}$$

The following are the IRC parts of the unpunctured and constrained ensembles obtained using $P_{\mathsf{HRC},2}$, where P_9 is based on PBD method and P_{10} is threshold-based:

$$P_{9,\mathsf{IRC}} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1\,1 \\ 0\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,1 \end{bmatrix}, \quad P_{10,\mathsf{IRC}} = \begin{bmatrix} 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0\,1\,0 \\ 1\,1\,0\,0\,0\,1\,0\,0 \\ 0\,1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,1 \\ 0\,1\,0\,1\,0\,1\,0\,0 \end{bmatrix} \tag{1.34}$$

Figs. 1.10 and 1.11 depict the minimum distance upper bounds and the iterative decoding thresholds of the corresponding ensembles. Unlike the results in Fig. 1.3, the constrained ensembles designed using the PBD method (P_6 and P_9) do not have thresholds similar to their threshold-based counterparts (resp. P_7 and P_{10}). In fact, the threshold difference, which is around 0.14dB across the range of design rates, is generally considered to be quite significant when comparing two codes designed for the BI-AWGNC.

We now present simulation results from the punctured ensembles P_6, P_7, and P_8 to demonstrate the effectiveness of the PBD method for low FER requirements. Although the

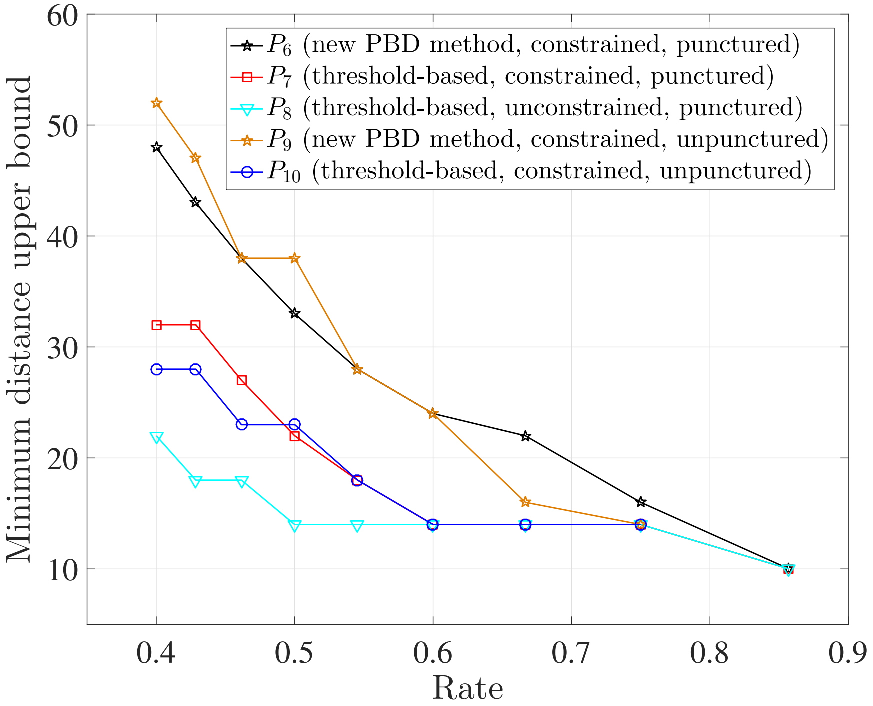

Figure 1.10: Comparison of minimum distance upper bounds of ensembles $P_i, 6 \leq i \leq 10$, obtained from HRC matrix $P_{\mathsf{HRC},2}$ in (1.31)

threshold differences in Fig. 1.11 are not promising, the simulated gap-to-capacity results in Fig. 1.12 show that PBD ensemble P_6 is the best-performing punctured ensemble considering all design rates, albeit with a worse, but negligible, performance at the lowest design rates when compared to ensemble P_7. Across results from both Section 1.7.1 and the current section, we observe that QC-LDPC code families designed using the PBD method possess the best slope in FER curves at the lowest FERs. Fig. 1.13 shows this to be true in the case of ensembles P_6, P_7, and P_8 at rate $6/15$. The figure also shows that the rate-$6/15$ code obtained from ensemble P_6 possesses the best empirical smallest codeword weight. This comparison suggests that designing threshold-based PBRL codes with the heuristic of constraining the row weights of P_{IRC} is a competitive alternative way to design short block-length PBRL codes for low FER requirements. We note that this specific constraint was not considered in the original work on PBRL codes [CVDW15], although the authors of [CVDW15] explored similar constraints involving individual edge weights. Overall, this comparison also validates our premise that the permanent bound identifies a better protomatrix when it comes to

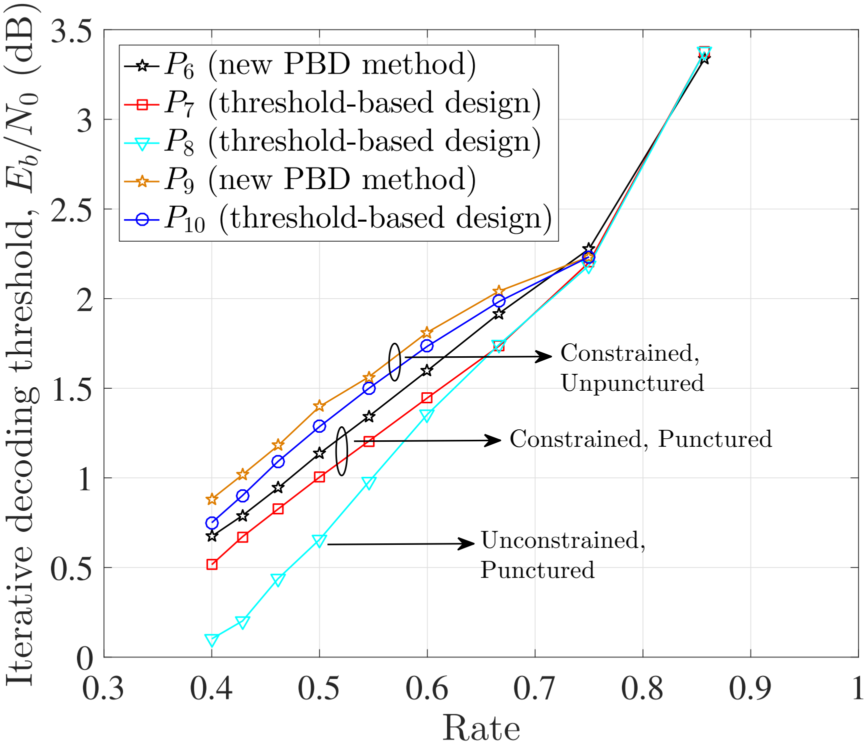

Figure 1.11: Iterative decoding thresholds over BI-AWGNC for ensembles $P_i, 6 \leq i \leq 10$, obtained from HRC matrix $P_{\mathsf{HRC},2}$

factors that affect low-FER performance at short block-lengths. Of course, designs that include both threshold and the permanent bound have the potential to be even more effective in those cases where the threshold difference is significant, but our initial results found that looking at the permanent bound alone can be sufficient for short block-lengths.

1.7.4 New Short Block-Length PBRL Codes for the 5G Standard

In order to show the effectiveness of our new design method for short block-length applications, we consider the short block-length PBRL codes for $k = 192$ information bits as

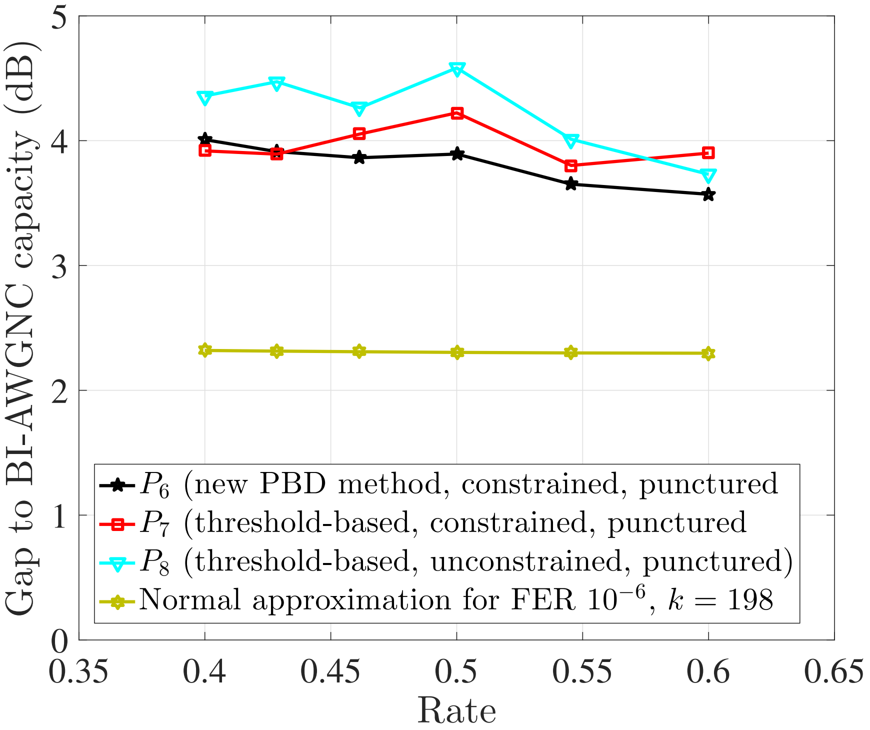

Figure 1.12: Comparison of gap to BI-AWGNC capacity at a frame error rate (FER) of 10^{-6} and information block size of $k = 198$ for RC families from ensembles P_6, P_7, and P_8 and the refined normal approximation of [PPV10]

specified in the 5G standard [Tec17]. The HRC matrix for this family of RC-LDPC codes is

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \tag{1.35}$$

where the first two variable nodes are punctured and the highest design rate is 6/8.

As the first two variable nodes are punctured, we used the full-complexity search procedure for the best set of rows based on the minimum distance upper bound until rate 6/15. Our design constrained each added row to the PBRL protomatrix to have exactly the same weight as the corresponding row of the PBRL protomatrix provided in the standard specification [Tec17]. Further, as for the connections between the punctured variable nodes and

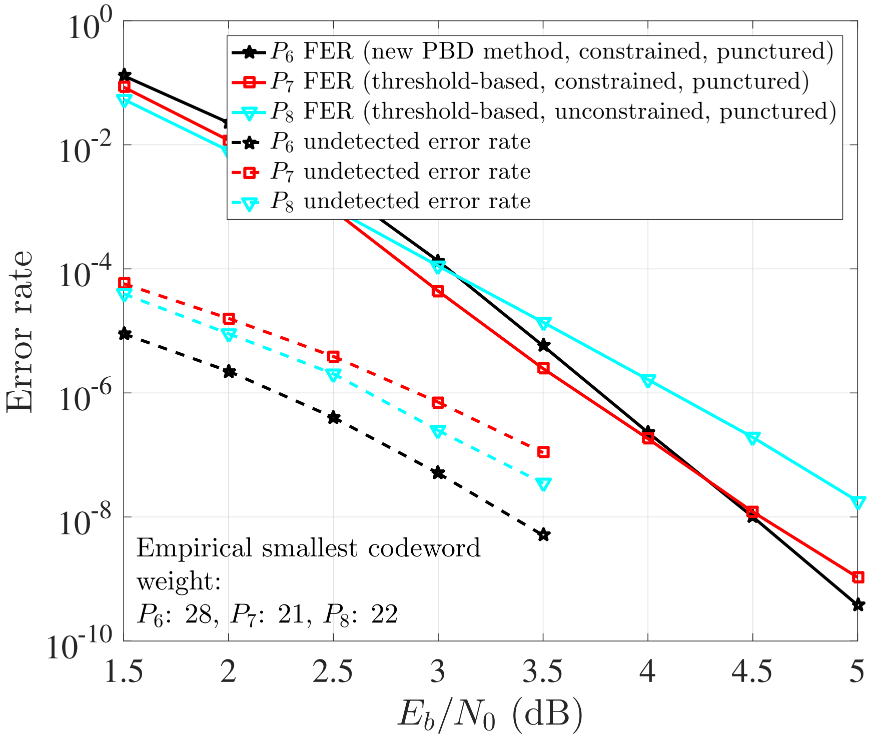

Figure 1.13: Comparison of FER and undetected error rate performance at rate 6/15 of QC-LDPC codes from ensembles P_6, P_7, and P_8

P_{IRC}, we retained the same connections as used in the standard. We then used our CPEG-ACE lifting algorithm to obtain a new set of $k = 192$ bits, QC PBRL codes with the same block-lengths as in the 5G standard.

Figs. 1.14 and 1.15 show simulation results over BI-AWGNC comparing the codes in the 5G standard and our newly obtained codes with the same parameters. Fig. 1.14 shows that the permanent-bound-based designs offer a better low-error-rate performance, once again confirming the premise of this chapter and the potential for the new design approach to be practically useful. A key point to note is that our design took only a few hours.

1.8 Concluding Remarks

This chapter proposed a new method to design PBRL QC-LDPC codes for short block-lengths and low FERs. The metric used in the design is an upper bound on the minimum distance of any QC-LDPC code that can be obtained from a protomatrix. By maximizing

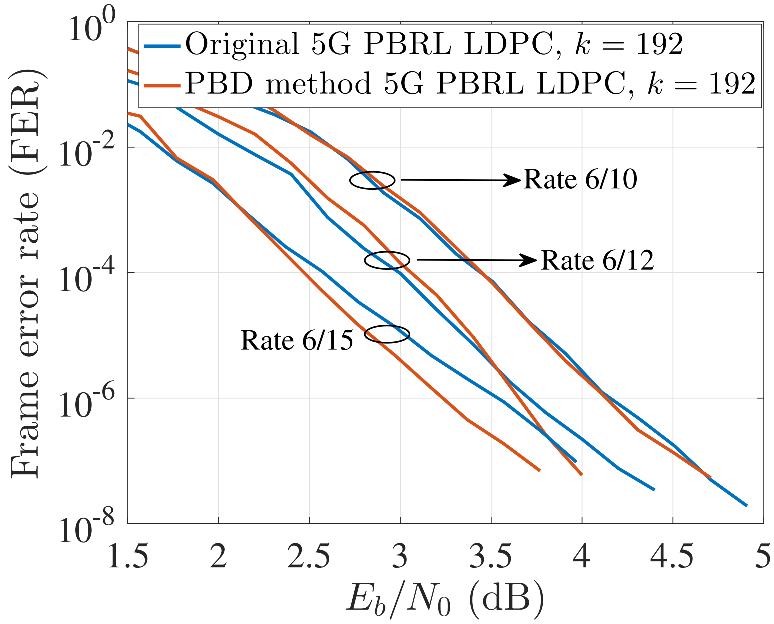

Figure 1.14: Comparison of FER performance of original 5G PRBL LDPC codes and the new PBD method 5G PBRL LDPC codes at $k = 192$ information bits

this upper bound at each design rate of the rate-compatible family of codes, the chapter obtained a significant improvement in the low-FER performance regime over PBRL codes designed according to the original method of optimizing the iterative decoding threshold. The chapter observed the combination of the original method of optimizing the iterative decoding threshold with a constraint on the row weights of the incremental check nodes to be an alternative way to design short block-length PBRL QC-LDPC codes. However, it is seen that the new design method produces QC-LDPC codes with better slope of FER performance at the lowest FERs when compared to the threshold-based methods.

Furthermore, the chapter identified a key reduction that is possible in the complexity of computing the minimum distance upper bounds. This reduction in complexity yielded a very low-complexity PBRL search procedure that has a constant run-time for each row added to lower the design rate. The constant complexity is in contrast to the original, threshold-based PBRL search procedure where the complexity of the search procedure grows with each new added row. Finally, the chapter established the equivalence between the newly proposed design method and an ILP and showed that the relaxation of this ILP is not exact.

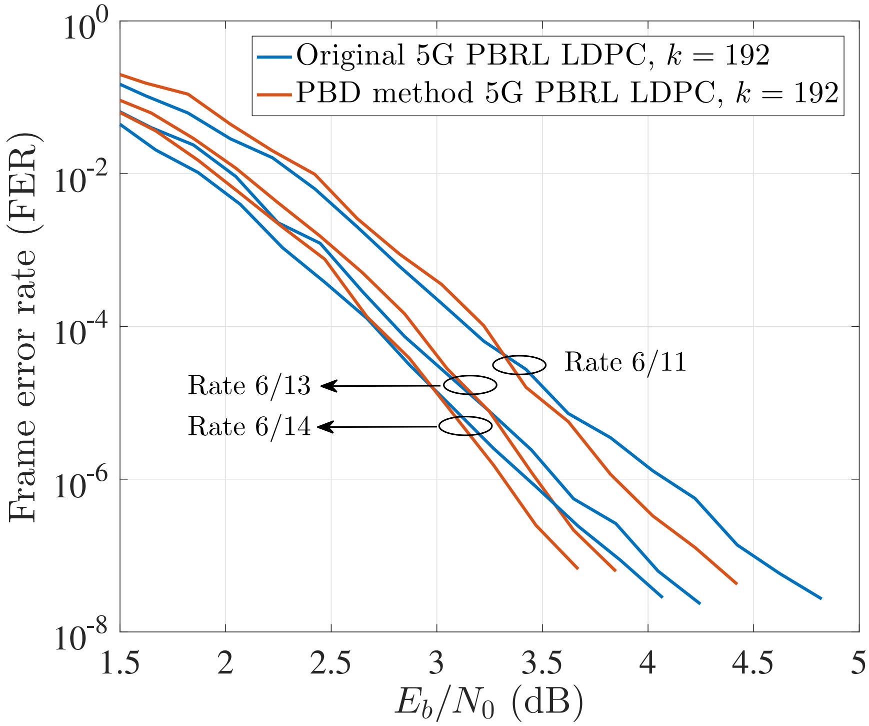

Figure 1.15: Comparison of FER performance of original 5G PRBL LDPC codes and the new PBD method 5G PBRL LDPC codes at $k = 192$ information bits

1.9 Appendix

1.9.1 Proof of Theorem 5

Proof. The cases when S contains the index of no punctured column or contains the index of exactly one punctured column are handled by the proofs of Theorems 3 and 4, respectively. Therefore, let us assume that S contains the indices of $n_{p_S} \geq 2$ punctured columns (out of $n_p > 1$). Let $P_{\mathsf{IR},\ell}$ denote the column in P_{IR} that contains its only non-zero entry, 1, in row ℓ. Also, assume that $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S\setminus i}\right) > 0$.

Let us consider a sub-matrix $P_{i^*} \triangleq P_{S \setminus i^*}$ with $i^* \in S \setminus \mathcal{P}$ such that $\mathsf{perm}\left(P_{i^*}\right) > 0$. This means that there exists a permutation σ^* such that $\prod_{j=1}^{n_c} P_{i^*}\left(\sigma^*(j), j\right) > 0$. If for such a permutation $\sigma^*(j) > n_{c_H}$ and $j > n_{p_S}$, then this means we can replace an index in S of an unpunctured column with an index of a column in P_{IR} to obtain a new set $\hat{S}$ such that $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S\setminus i}\right) \geq \sum_{i \in \hat{S} \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}\setminus i}\right) > 0$, unless the latter is already present in S. In the same way, we can now iteratively obtain newer sets until we arrive at a set $\hat{S}_1$ such that the following is true: For every sub-matrix $P_{\hat{S}_1 \setminus i}$, $i \in \hat{S}_1 \setminus \mathcal{P}$ with $\mathsf{perm}\left(P_{\hat{S}_1 \setminus i}\right) > 0$, for every permutation σ such that $\prod_{j=1}^{n_c} P_{\hat{S}_1 \setminus i}\left(\sigma(j), j\right) > 0$, the conditions $P_{\hat{S}_1 \setminus i}\left(\sigma(j), j\right) > 0$, $j > n_{p_S}$, and $\sigma(j) > n_{c_H}$ are satisfied only if $P_{\mathsf{IR},\sigma(j)}$ is already indexed in set $\hat{S}_1$.

Now, if $\hat{S}_1$ contains the indices of all the columns of P_{IR}, then the proof is complete as $\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) \geq \sum_{i \in \hat{S}_1 \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}_1 \setminus i}\right) > 0$. If that is not the case, then we proceed to do further replacements, but using the properties of $\hat{S}_1$ and the assumptions in the statement of the theorem. Note that $\hat{S}_1$ contains the indices of all the punctured columns that were indexed in S.

Consider a column P_{IR,ℓ^*} whose index is not contained in $\hat{S}_1$. For any sub-matrix $P_{\hat{S}_1 \setminus i}$, $i \in \hat{S}_1 \setminus \mathcal{P}$ with $\mathsf{perm}\left(P_{\hat{S}_1 \setminus i}\right) > 0$, every permutation σ with a non-zero product $\prod_{j=1}^{n_c} P_{\hat{S}_1 \setminus i}\left(\sigma(j), j\right)$ has $\sigma(j) = \ell^*$ only when $j \leq n_{p_S}$. Pick any such sub-matrix and one such permutation σ^* for that sub-matrix. There is only one $j^* \leq n_{p_S}$ with $P_{\hat{S}_1 \setminus i}\left(\sigma^*(j^*), j^*\right) > 0$ and $\sigma^*(j^*) = \ell^*$ as we have assumed that only one punctured column has a non-zero entry in every row of P_{IRC}. Note that $j^* \leq n_{p_S}$ is the local index (i.e. within the sub-matrix) of a column in the chosen $P_{\hat{S}_1 \setminus i}$ that is originally a punctured column in P. Let the index of this punctured column in P be i_1^*.

Now, similar to Case 2 of the proof of Theorem 4, assume that the matrix $P_{\hat{S}_1 \setminus i_1^*}$, which is not part of $\sum_{i \in \hat{S}_1 \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}_1 \setminus i}\right)$, has a positive permanent. As the HRC part has a positive design rate less than 1, $n_{p_S} \leq n_p < n_{c_H}$. Then, with the assumptions of the theorem that only one punctured column has a non-zero entry in every row of P_{IRC}, that all punctured columns have all entries in their first n_{c_H} rows as non-zero integers, and the fact that $n_{p_S} \leq n_p < n_{c_H}$, we can find a sub-matrix $P_{\hat{S}_1 \setminus i}, i \in \hat{S}_1 \setminus \mathcal{P}$ with a positive permanent such that there exists a permutation σ with a non-zero product $\prod_{j=1}^{n_c} P_{\hat{S}_1 \setminus i}\left(\sigma(j), j\right)$ and $\sigma(j) = \ell^* > n_{c_H}$ for some $j > n_{p_S}$. But, $\hat{S}_1$ is a set that cannot have this property and therefore our assumption that $P_{\hat{S}_1 \setminus i_1^*}$ has a non-zero permanent is incorrect.

Now, with the assumption that $P_{\hat{S}_1 \setminus i_1^*}$ can only have a permanent that is equal to zero, we can replace $i_1^* \in S$ with the index of P_{IR,ℓ^*} to obtain a new set $\hat{S}_2$ that has one fewer index missing of columns in the incremental redundancy part of P such that

$$\sum_{i \in \hat{S}_1 \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}_1 \setminus i}\right) \geq \sum_{i \in \hat{S}_2 \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}_2 \setminus i}\right) > 0. \tag{1.36}$$

Now, $\hat{S}_2$ is another set of $n_c + 1$ column indices that has the relationship with S that

$$\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) \geq \sum_{i \in \hat{S}_2 \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}_2 \setminus i}\right) > 0. \tag{1.37}$$

If $\hat{S}_2$ contains the indices of all the columns of P_{IR}, then the proof is complete. Otherwise we can follow through the same steps that we have applied to set S so far, and by iteratively repeating these steps we can arrive at a set $\hat{S}^*$ that contains the indices of all the columns of P_{IR} such that

$$\sum_{i \in S \setminus \mathcal{P}} \mathsf{perm}\left(P_{S \setminus i}\right) \geq \sum_{i \in \hat{S}^* \setminus \mathcal{P}} \mathsf{perm}\left(P_{\hat{S}^* \setminus i}\right) > 0. \tag{1.38}$$

□

Acknowledgment

The work that resulted in this chapter was supported by NSF under grant CIF-1618272. Research was carried out in part at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with NASA. We are grateful to Fabian Steiner of TU Munich for assistance with simulation results in Section 1.7.4.

CHAPTER 2

Linear Rate-Compatible Codes with Degree-1 Extending Variable Nodes Under Iterative Decoding

2.1 Introduction and Notation

Chapter 1 of this dissertation introduced a new design method for PBRL codes. This chapter[1] studies decoding properties of PBRL codes and questions whether the extending structure based on degree-1 variable nodes is optimal or not in the threshold sense. Though the objective of this chapter is to focus on PBRL codes at its core, the discussion applies to a more general class of codes; we first introduce this general class of codes in the following.

2.1.1 Rate-Compatible Extension with Degree-1 Variable Nodes

Let $H_{m\times n}$ denote the parity-check matrix of a binary linear code of length n. The m rows of the parity-check matrix are called the check nodes and the n columns are called the variable nodes, as defined usually in literature on LDPC codes (See [RU01] and Chapter 1). The degree of a node is the number of edges emanating from a node in the Tanner graph representation of the parity-check matrix. Our focus in this chapter is on LDPC codes [Gal63], although some of the results in this chapter are applicable for any linear code with the structured considered in this chapter.

[1] Almost all of this chapter was presented at the 2018 IEEE International Symposium on Information Theory (ISIT) [RWD18].

We consider linear codes that possess a special structure in their parity-check matrix H. This structure is essentially a generalization of the PBRL structure to binary linear codes. Specifically, let H_{HRC} denote the parity-check matrix of the code of the highest rate in an RC code with extension. Let every variable node that is transmitted to lower the rate of the code be of degree 1. The parity-check matrix of the RC code at subsequent lower rates has the following general structure:

$$H = \begin{bmatrix} H_{\mathsf{HRC}} & 0 \\ H_{\mathsf{IRC}} & I \end{bmatrix}_{m \times n} \tag{2.1}$$

Here, 0 and I represent the all-zeros and identity matrices of appropriate sizes. Let the size of H_{HRC} be $m_H \times n_H$. IR variable node $i, n_H + 1 \leq i \leq n$, is generated as the linear combination (XOR) of HRC variable nodes that check node $m_H + i$ is connected to in H. The sub-matrix H_{IRC} represents the connections involved in generating the IR symbols, where IRC stands for "incremental redundancy code". Let us call any parity-check matrix with the structure in (2.1) a *Raptor-like parity-check matrix*, or simply a *raptor-like (RL) matrix.* For simplicity, we use the term “IRC check node” to describe a check node connected to any IR variable node.

The variable nodes in the initial transmission (the first n_H columns of H) could also contain punctured nodes. The rate r of a parity-check matrix of size $m \times n$ with n_0 punctured variable nodes satisfies $r \geq (n - m)/(n - n_0)$, with equality whenever the matrix is full-rank.

Two well-known classes of codes with RL matrices are Shokrollahi’s Raptor codes [Sho06] and PBRL codes. Raptor codes with a specific degree distribution to form random connections in H_{IRC} and a good-enough H_{HRC} achieve the capacity of any binary erasure channel (BEC) [Sho06]. These codes also have good performance over any memoryless symmetric channel [ES06]. PBRL codes, designed for simplicity and low encoding complexity for physical layer transmission, have been shown to have an excellent iterative decoding threshold and performance over BEC, binary symmetric channel (BSC), and binary-input additive white-

Gaussian noise channel (BI-AWGNC). PBRL codes have a good frame error rate (FER) performance across the entire family of design rates over both short and long block-lengths: see [CVDW15] and Chapter 1 for BI-AWGNC and [VDW15] for BEC.

2.1.2 Iterative Decoders and Convergence

In this chapter, we consider transmission over the BEC, BSC, and BI-AWGNC with iterative decoding at the receiver (See [RU01] for iterative decoding). The input to the iterative decoder is the following set of log-likelihood ratios (LLRs):

$$\hat{L}_{v_i} = \log\left(\frac{\mathbb{P}\left[v_i = 0 \mid y_i\right]}{\mathbb{P}\left[v_i = 1 \mid y_i\right]}\right), \quad 1 \leq i \leq n, \tag{2.2}$$

where v_i is the input to the channel and y_i is the output of the channel corresponding to v_i.

For the BEC, the iterative decoder that we consider is the peeling decoder [LMSS01], for which we do not need the description in terms of the LLR values. The output from a BEC is either the transmitted bit or an erasure. A peeling decoder process performs the following operation repeatedly: If there is a check node with all but one known variable node value, set the missing variable node to be the XOR of its known message bits. The process terminates if there are no such check nodes.

For decoders that operate based on LLR values, we consider two specific methods: belief propagation (BP) and min-sum decoding [RL09]. Let the message during iteration ℓ from variable node $v_i, 1 \leq i \leq n$, to its neighboring check node $c_j, c_j \in \mathcal{N}_{v_i}$, be denoted $L^{\ell}_{v_i \to c_j}$, where $\mathcal{N}_{v_i}$ denotes the set of neighboring checks of v_i. Similarly, let the message during iteration ℓ from check node $c_j, 1 \leq j \leq m$, to its neighboring variable node v_i be denoted $L^{\ell}_{c_j \to v_i}, v_i \in \mathcal{N}_{c_j}$. Every iteration $\ell \geq 1$ proceeds with check nodes processing and transmitting messages and ends with variable nodes processing and transmitting messages. The initialization messages are

$$L^{0}_{v_i \to c_j} = \hat{L}_{v_i}, \quad c_j \in \mathcal{N}_{v_i}, \quad 1 \leq i \leq n\,. \tag{2.3}$$

For both BP and min-sum decoding, the variable-to-check update function is given as

$$L^{\ell}_{v_i \to c_j} = \sum_{c_s \in \mathcal{N}_{v_i} \setminus c_j} L^{\ell}_{c_s \to v_i} + \hat{L}_{v_i}\,. \tag{2.4}$$

Let $\phi(x)$ be defined as

$$\phi(x) = \log\left(\frac{e^x + 1}{e^x - 1}\right), \quad x \in (0, \infty)\,. \tag{2.5}$$

Note that $\phi(x)$ takes values in $(0, \infty)$ and satisfies $\phi(x) = \phi^{-1}(x)$. In the case of BP, the check-to-variable update function is given as follows:

$$\begin{aligned} L^{\ell}_{c_j \to v_i} = & \prod_{v_s \in \mathcal{N}_{c_j} \setminus v_i} \operatorname{sign}\left(L^{\ell-1}_{v_s \to c_j}\right) \cdot \\ & \phi\left(\sum_{v_s \in \mathcal{N}_{c_j} \setminus v_i} \phi\left(|L^{\ell-1}_{v_s \to c_j}|\right)\right) \end{aligned} \tag{2.6}$$

The check-to-variable update function of the min-sum decoder is an approximation of (2.6):

$$L^{\ell}_{c_j \to v_i} = \prod_{v_s \in \mathcal{N}_{c_j} \setminus v_i} \operatorname{sign}\left(L^{\ell-1}_{v_s \to c_j}\right) \cdot \min_{v_s \in \mathcal{N}_{c_j} \setminus v_i} |L^{\ell-1}_{v_s \to c_j}| \tag{2.7}$$

The convergence of the iterative decoder in a finite number of iterations, in practical implementations, is checked the following way: At the end of each iteration $\ell \geq 1$, the variable nodes compute their overall LLR value as

$$L^{\ell}_{v_i} = \sum_{c_j \in \mathcal{N}_{v_i}} L^{\ell}_{c_j \to v_i} + \hat{L}_{v_i}\,. \tag{2.8}$$

Each variable node then makes the following hard decision:

$$\hat{X}_i = \begin{cases} 0, & \text{if } L_{v_i}^{\ell} \geq 0, \\ 1, & \text{if } L_{v_i}^{\ell} < 0. \end{cases} \tag{2.9}$$

Definition 2 (Practical convergence criterion)**.** *The decoder is said to have converged at the end of iteration ℓ if the hard decisions $\hat{X}_i, 1 \leq i \leq n$, identify a codeword of $H_{m\times n}$.*

Other Notation: We use ∞_i as shorthand to denote a constant value of ±∞ associated with an index i.

2.1.3 Organization

Section 2.2 discusses convergence properties of RL codes under iterative decoding and provides results from simulations of PBRL codes. Section 2.3 discusses the closeness of iterative decoding thresholds of PBRL ensembles to thresholds of protograph-based RC extension using the optimal degrees for extending nodes. Section 2.4 concludes the chapter.

2.2 Convergence in Iterative Decoding of Raptor-Like Matrices

In this section, we discuss convergence results for iterative decoding of RL matrices and provide simulation data from PBRL codes to substantiate these results. The results of Section 2.2.1 show that the practical convergence criterion (Definition 2) need only check whether HRC variable nodes have converged to a codeword. Since the message can be recovered using the HRC symbols, it is sufficient to recover the HRC codeword. Simulation results in Section 2.2.2 show that checking only the HRC variable nodes for convergence in a practical setting does not lead to a deterioration in FER performance of PBRL codes.

As an initial observation, if an RL matrix is used over the BEC with a peeling decoder, it is apparent that if the HRC variable nodes have been recovered then the IR variable nodes are also recovered eventually by the peeling process. If the iterative decoder for BEC is implemented analogous to LLR-based BP, then it is clear that once the HRC nodes are recovered with a ±∞ value of overall LLR $L_{v_i}^{\ell}$, any unrecovered IR node will be recovered in the next iteration.

2.2.1 Convergence

The first result is an observation regarding the magnitude of messages from IRC check nodes to HRC variable nodes.

Lemma 4. *Consider the BP or the min-sum decoder. Let c_j be a check node with neighbors $v_1, v_2, \ldots$. If v_1 is of degree 1, then the outgoing message from c_j to any of its neighbors other than v_1 during any iteration ℓ is bounded in magnitude.*

Proof. In the case of the check-to-variable update equation (2.6) in BP, since $\phi(x)$ is positive and monotonically decreasing with x, for any $v_i \neq v_1$ and for any $\ell \geq 1$

$$\phi\left(\sum_{v_s \in \mathcal{N}_{c_j} \setminus v_i} \phi\left(|L_{v_s \to c_j}^{\ell-1}|\right)\right) \leq \phi\left(\phi\left(|L_{v_1 \to c_j}^{\ell-1}|\right)\right) \tag{2.10}$$

$$= |L_{v_1 \to c_j}^{\ell-1}| = |\hat{L}_{v_1}|. \tag{2.11}$$

The proof for min-sum decoding is similar. □

Using Lemma 4, we now show a result in the infinite iterations case for convergence as defined in Definition 2.

Theorem 6. *Let the HRC variable nodes of an RL matrix under BP or min-sum decoding satisfy $\lim_{\ell \to \infty} L_{v_i}^{\ell} = \infty_i, 1 \leq i \leq n_H$, for some set of ∞_i values for $1 \leq i \leq n_H$. Then, the IR variable nodes will also satisfy $\lim_{\ell \to \infty} L_{v_i}^{\ell} = \infty_i, n_H + 1 \leq i \leq n$, for a set of ∞_i values for $n_H + 1 \leq i \leq n$.*

Proof. The IRC check nodes are limited in the magnitude of messages that they can send in any iteration to HRC variable nodes (Lemma 4). Because of the assumption that $\lim_{\ell\to\infty} L^{\ell}_{v_i} = \infty_i, 1 \le i \le n_H$, the outgoing messages to the IRC check nodes from HRC variable nodes also satisfy $\lim_{\ell\to\infty} L^{\ell}_{v_i\to c_j} = \infty_i, c_j \in \mathcal{N}_{v_i}, c_j$ an IRC check node, $1 \le i \le n_H$. Therefore, the magnitude of messages from IRC check nodes to IR variable nodes under both BP and min-sum decoding satisfy $\lim_{\ell\to\infty} L^{\ell}_{c_j\to v_i} = \infty_i, c_j$ an IRC check node, $n_H + 1 \le i \le n, v_i \in \mathcal{N}_{c_j}$, for some set of ∞_i values for $n_H+1 \le i \le n$. This implies that $\lim_{\ell\to\infty} L^{\ell}_{v_i} = \infty_i, n_H+1 \le i \le n$.

In fact, the sign of $\infty_i, n_H + 1 \le i \le n$, can be determined as the product of the signs of ∞_i that the variable nodes that IR node i is the XOR of. Therefore, if the HRC variable nodes converge in the sense stated in the theorem to a codeword, then the IR nodes will converge to satisfy the IRC check nodes. □

The convergence to a codeword in the sense that all hard decisions as defined in (2.9) satisfy all parity checks in the graph is used as a stopping criterion in practical decoding. From a theoretical perspective, there is another type of convergence that is usually defined in obtaining the iterative decoding threshold of a degree distribution of an LDPC code via density evolution [RU01]. Assuming that the all-zeros codeword is transmitted, the latter criterion requires $L^{\ell}_{v_i\to c_j} \to +\infty$ for all variable nodes of an infinite Tanner graph. This form of convergence can never be satisfied for variable nodes of degree 1 as we can see that $L^{\ell}_{v_i\to c_j} = \hat{L}_{v_i}, \forall \ell$. The following observation combines the two forms of convergence to show a result similar to Theorem 6:

Theorem 7. *Let the* HRC *variable nodes of an RL matrix under iterative BP or min-sum decoding satisfy* $\lim_{\ell\to\infty} L^{\ell}_{v_i\to c_j} = \infty_i$ *for all neighboring* HRC *check nodes* c_j *of* v_i *for some set of* ∞_i *values for* $1 \le i \le n_H$*. Then, the overall LLR values of the IR variable nodes will satisfy* $\lim_{\ell\to\infty} L^{\ell}_{v_i} = \infty_i, n_H + 1 \le i \le n$*, for some* ∞_i *values.*

Proof. Consider an HRC variable node v_i. The condition $\lim_{\ell\to\infty} L^{\ell}_{v_i\to c_j} = \infty_i$ for all neighboring HRC check nodes c_j of v_i means that for every such check node c_j

$$\lim_{\ell \to \infty} \sum_{c_s \in \mathcal{N}_{v_i} \setminus c_j} L^{\ell}_{c_s \to v_i} = \infty_i. \tag{2.12}$$

For v_i, denote the set of neighboring HRC check nodes $\mathcal{N}^{H}_{v_i}$. Now, since the messages from the IRC check nodes to the HRC variable nodes are bounded in magnitude,

$$\lim_{\ell \to \infty} \sum_{c_s \in \mathcal{N}^{H}_{v_i} \setminus c_j} L^{\ell}_{c_s \to v_i} = \infty_i. \tag{2.13}$$

Let $|\mathcal{N}^{H}_{v_i}| \triangleq d^{H}_{v_i}$, the degree of variable node v_i in the highest rate code. Since

$$\begin{aligned} L^{\ell}_{v_i} = \frac{\sum_{c_j \in \mathcal{N}^{H}_{v_i}} \sum_{c_s \in \mathcal{N}^{H}_{v_i} \setminus c_j} L^{\ell}_{c_s \to v_i}}{d^{H}_{v_i} - 1} \\ + \sum_{c_j \in \mathcal{N}_{v_i} \setminus \mathcal{N}^{H}_{v_i}} L^{\ell}_{c_j \to v_i} + \hat{L}_{v_i}, \end{aligned} \tag{2.14}$$

we can see that $L^{\ell}_{v_i} \to \infty_i$ as $\ell \to \infty$. By now applying Theorem 6, we arrive at the result. □

Theorem 7, in fact, shows the following general result:

Corollary 1. *Let the non-degree-1 variable nodes of a parity-check matrix under iterative BP or min-sum decoding satisfy* $\lim_{\ell \to \infty} L^{\ell}_{v_i \to c_j} = \infty_i$ *for all neighboring check nodes* c_j *of* v_i *that do not have as neighbors any degree-1 variable node, for some set of* ∞_i *values for* $1 \le i \le n$. *Then, the overall LLR values of all the variable nodes will satisfy* $\lim_{\ell \to \infty} L^{\ell}_{v_i} = \infty_i, 1 \le i \le n$, *for a set of* ∞_i *values.*

The results of this section have shown that, in the case of infinite iterations it suffices to consider the convergence of only the HRC variable nodes. We end the discussion by providing a further thought on why it may be sufficient to be concerned only with the convergence of the HRC variable nodes from the perspective of reducing the average number of iterations in the implementation of a decoder. Let A_i be the event that a codeword is selected by the hard decisions of (2.9) satisfying all (i.e. both HRC and IRC) check nodes and that variable node i is decoded in error. It can be seen that

$$\left\{ \bigcup_{i=n_H+1}^{n} A_i \right\} \Longrightarrow \left\{ \bigcup_{i=1}^{n_H} A_i \right\}. \tag{2.15}$$

That is, if a codeword satisfies all (i.e. both HRC and IRC) parity checks and there is an error in the IR variable nodes, then there must also be an error in the HRC variable nodes. Therefore, if the HRC variable nodes satisfy the HRC parity checks in some iteration ℓ and those HRC variable nodes identify the correct HRC codeword, then the IR nodes will be correct if all (i.e. both HRC and IRC) parities check according to the hard decision of (2.9) in some later iteration $\ell + i$ and the HRC hard decisions remain as they were in iteration ℓ.

This means that for those RL matrices with a very low probability of converging to the wrong HRC codeword overall, one would save on the number of iterations performed while decoding the received LLR values if the implementation does not wait until all the parities check and accepts the HRC hard decisions if they converge to a codeword according to the hard decisions of (2.9). As an aside, note that it is also true that if there are errors in the HRC variable nodes upon decoding to a codeword by checking all parities of H, then it does not necessarily mean that there is an error in the IR nodes.

2.2.2 Convergence and Frame Error Rates of PBRL Codes

We consider the specific PBRL ensemble referred to as P_3 in Chapter 1 over BI-AWGNC for the results presented in this section, but we observe the same behavior with other PBRL ensembles as well. Ensemble P_3 has an HRC protograph P_{HRC} of rate 6/7 and an overall protograph P with the lowest rate of 6/15. The design rates of the ensemble are $6/(7+i), 0 \leq i \leq 8$.

Fig. 2.1 shows the FER under BP decoding of a code at rate 6/9 from ensemble P_3. A

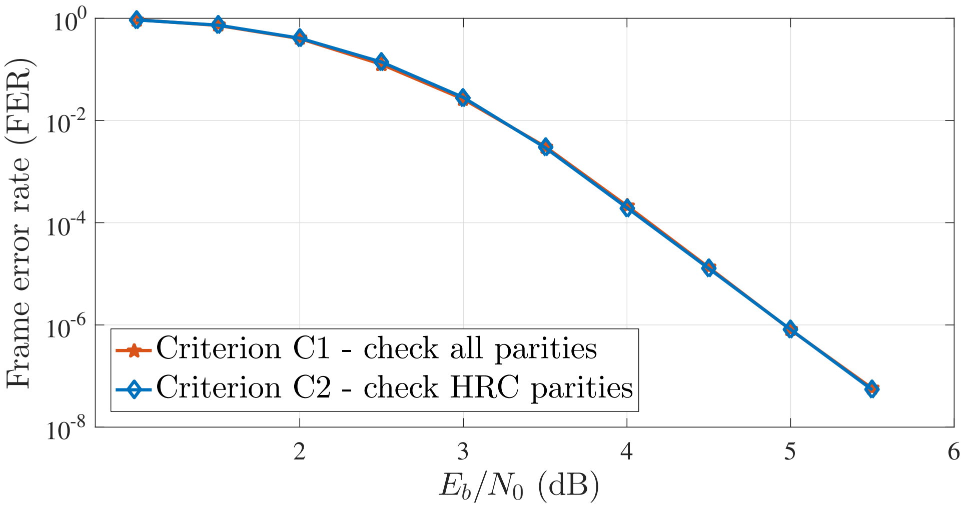

Figure 2.1: Comparison of FER between criteria C1 and C2 for a rate-6/9 PBRL code from ensemble referred to as P_3 in Chapter 1

frame is said to have been decoded in error if either the decoder converges to an incorrect codeword or fails to converge to a codeword within a set maximum number of iterations. With a maximum of 100 iterations, the figure shows the performance of the code under two convergence criteria considered after each iteration. The first criterion, called C1, accepts a set of hard decisions only if all the parities of the code check and the second, called C2, requires only the HRC check nodes to be satisfied to accept a decoding decision. We see that there is no difference in the performance between the two criteria. This figure, and performances we have observed with other PBRL codes, show that it may suffice to consider the convergence of only the HRC variable nodes after each iteration.

Fig. 2.2 shows the histogram of number of iterations required for convergence to a codeword under the criteria C1 and C2 for the same code at rate 6/9 and an E_b/N_0 of 4dB. The figure shows only iterations 1 through 10 on the x-axis for clarity of presentation. At E_b/N_0 of 4dB, the average number of iterations was observed to be 3.67 under criterion C1 and 3.30 under C2. As another example, the code at rate 6/12 of the ensemble converged in an average of 5.39 iterations under C1 and 4.44 iterations under C2 at an E_b/N_0 of 3dB. The savings in the average number of iterations presented here for the two codes are small when compared to the maximum of 100 allowed iterations. However, even a small improvement

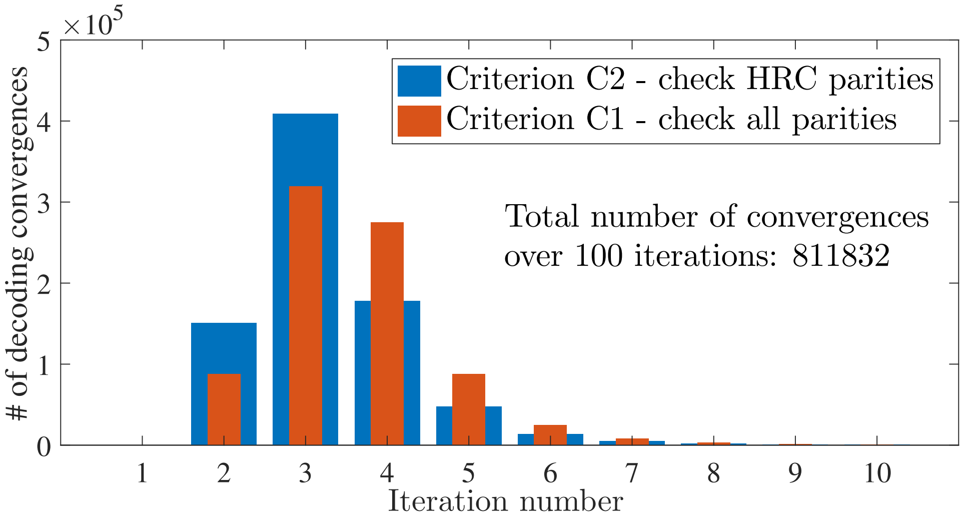

Figure 2.2: Number of decoding convergences to a codeword with criteria C1 and C2 for the code whose performance is shown in Fig. 2.1; $E_b/N_0 = 4dB$

is crucial when it comes to high-throughput decoder implementations in practice where the maximum number of iterations is typically a number such as 10.

Complexity Savings: Convergence criterion C2 has two advantages over C1. One benefit is that we do not need to compute the overall LLR values L_{v_i} of the IR variable nodes after each iteration. The second benefit is the increased throughput due to fewer average number of iterations performed at the decoder with no deterioration in the performance of the code.

2.3 Comparison of Thresholds of PBRL Protographs to Optimally Extended Protographs

A parity-check matrix with a lower-triangular sub-matrix instead of the identity matrix in an RL code defined by (2.1) can be transformed into an RL matrix by elementary row operations of the last $m - m_H$ rows of H. Therefore, when it comes to maximum-likelihood (ML) decoding over a binary-input memoryless symmetric output channel, removing the requirement of the identity matrix structure does not provide any benefit while constructing an extended RC code. That is, an RC code with extending variable nodes of higher degrees is equivalent under ML decoding and in the sub-space spanned by the code to an RL code

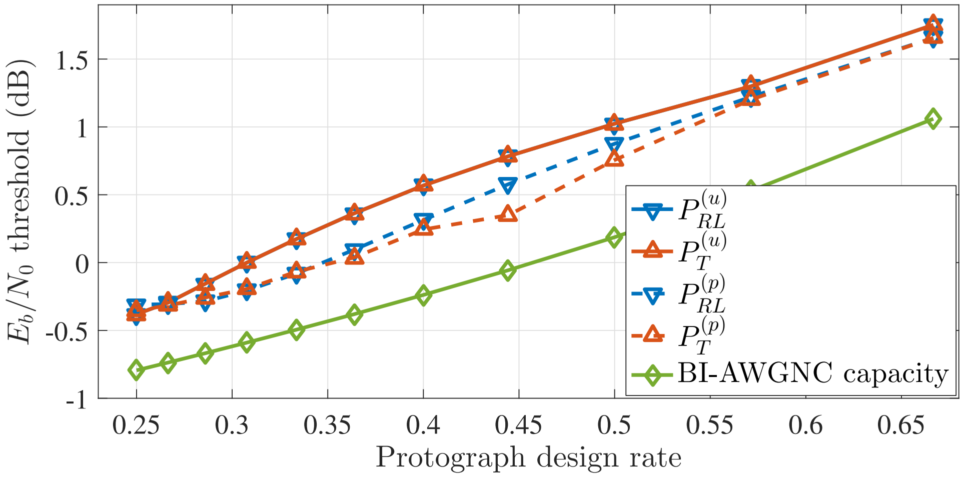

Figure 2.3: Iterative decoding threshold over BI-AWGNC: threshold-optimal PBRL extension vs. threshold-optimal RC extension of P_{HRC} in (2.16)

(with only degree-1 extending variable nodes) with a possibly different H_{IRC}.

In this section, we explore whether allowing a lower-triangular sub-matrix in the extended RC protograph (and hence in the lifted code) provides significant benefits for a protograph ensemble in terms of the iterative decoding threshold. The PBRL design procedure, as proposed in [CVDW15], obtains a PBRL rate-compatible protograph the following way: Given an HRC protograph, the objective is to add rows to the protograph to lower the rate by keeping all previously obtained rows fixed. This is done while optimizing the iterative decoding threshold of the protograph over the channel under consideration.

Here, we begin with the HRC protograph

$$P_{\mathsf{HRC}} = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 \end{bmatrix} \tag{2.16}$$

and optimize the entries of P_{IRC} by considering values of 0 and 1. The HRC protograph (2.16) has a design rate of 4/6 when there are no punctured variable nodes and has a design rate of 4/5 when the first column is punctured; we consider both cases. We assume BI-AWGNC and BSC here; the results for BEC are similar and are omitted. For each channel, we obtain the following optimized protograph ensembles: $P_{RL}^{(p)}, P_{RL}^{(u)}, P_{T}^{(p)}, P_{T}^{(u)}$, where the

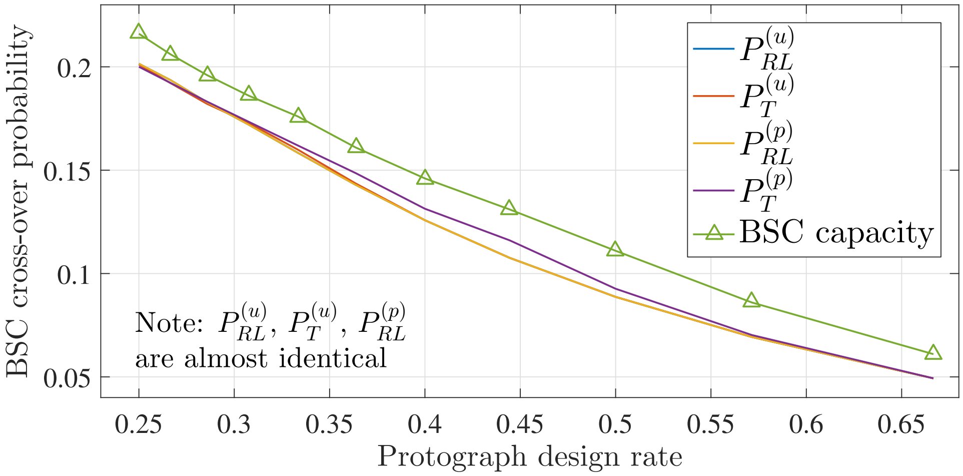

Figure 2.4: Iterative decoding threshold over BSC: threshold-optimal PBRL extension vs. threshold-optimal RC extension of P_{HRC} in (2.16)

subscript identifies whether the protograph is raptor-like (RL) or lower-triangular (T) and the superscript indicates whether the first column is punctured (p) or unpunctured (u).

Figs. 2.3 and 2.4 depict the resulting thresholds. The results show that when the protograph (2.16) has no punctured variable nodes there is no benefit (in terms of threshold) in allowing extending variable nodes with higher degree. With the presence of a punctured variable node there is some benefit, but the threshold values of $P_{RL}^{(p)}$ and $P_{T}^{(p)}$ are close to each other. The figures show that there is a rate at which the difference in thresholds is noticeable. We observed that if we increased the size of the HRC protograph by using more variable nodes and the same number of check nodes, the rate at which this noticeable difference occurs increases and the thresholds of $P_{RL}^{(p)}$ and $P_{T}^{(p)}$ are close at the other rates.

2.4 Concluding Remarks

This chapter discussed properties of raptor-like codes. We showed that the practical convergence criterion after each iteration need only check whether the HRC variable nodes have converged. We provided evidence from PBRL codes that such a decoder shows no deterioration in performance in terms of the frame error rate, converges in a fewer iterations, and has a lower complexity. We also presented iterative decoding threshold results that demonstrate that PBRL codes are very close in terms threshold to extension using the threshold-optimal lower-triangular connections for the incremental variable nodes of the protograph.

Acknowledgment

The work that resulted in this chapter was supported in part by NSF under grant CIF-1618272. Research was carried out in part at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with NASA. We acknowledge Kasra Vakilinia's experimental work on decoding PBRL codes, which inspired parts of this chapter.

CHAPTER 3

Universal Rate-Compatible LDPC Code Families for Any Increment Ordering

Recall from Chapters 1 and 2 that an RC code family supports successively lower code rates by sending specific increments of additional redundancy at each rate. That is, the order of the increments is fixed. However, in some multi-hop communication systems and also in recently proposed inter-frame coding, the order in which the decoder of the RC code receives the increments is not predetermined. A different ordering of the increments at the decoder may change the codes of various rates.

This chapter[1] seeks RC codes that are universally good over all increment orderings. We call RC codes satisfying this requirement *universal for any increment ordering* (UIO) codes. It turns out that PBRL codes (See Chapters 1 and 2) are well suited to tackle this requirement. We design PBRL code ensembles for UIO codes using protograph thresholds as components of two design metrics. One metric seeks codes that, at each code rate, have exactly the same frame error rate for all increment orderings. The other metric sacrifices strictly identical performance for every ordering to seek codes that achieve the best possible throughput in a variable-length setting with random increment ordering, as would occur with inter-frame coding. Simulation results of UIO-PBRL codes from the new ensembles show that our designs satisfy the two metrics.

[1] Almost all of this chapter was presented at the 2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC) [RVDW16].

3.1 Introduction

This chapter, which continues the exploration PBRL codes, is motivated by Zeineddine and Mansour's recently proposed *inter-frame coding* [ZM16], which works as follows: A certain number of message packets is to be transmitted. They are each coded separately using an RC code and the highest-rate parts are sent out. Along with this, linear combinations of increments are also transmitted, where each of the increments for a combination comes from the RC code for a different packet. At the receiver, whenever a channel packet is decoded successfully, all increments of its RC code become known and are used to reveal (from the linear combinations) new increments for the remaining packets. These new increments lower the rate of their corresponding RC-coded packets, hopefully allowing some of them to be decoded so that the process iterates and recovers all the message packets.

The decoder of the RC codes in an inter-frame code [ZM16] is not guaranteed to see a specific ordering of its increments at each code rate. Therefore, in order to support a practical implementation of the scheme, which was not considered in [ZM16], we need an RC code that performs well over all increment orderings. We call RC codes satisfying this requirement *universal for any increment ordering* (UIO) codes. It was noted in [ZM16] that such codes had yet to be investigated in literature.

As an aside, [ZM16] also assumes an unrealistic distribution of successful decodings as a function of rate-compatible code block-length. The work of Wang et al. [WRW17] builds on the work [ZM16] and provides a practical inter-frame code construction. The authors of [WRW17] use a convolutional code as the highest-rate code for all component codes and increments of the component codes are linearly combined using a degree distribution designed carefully using differential evolution. The authors of [WRW17] incorporate the actual observed empirical distribution of success decodings as a function of the rate-compatible convolutional code block-length.

This chapter designs UIO codes with PBRL LDPC structure via two design metrics.

The first metric seeks codes that have, at each rate, the same FER performance for every increment ordering. This metric reflects the key assumption of the throughput analysis of inter-frame coding in [ZM16] that at each rate the RC codes in an inter-frame code will perform exactly the same for any increment ordering. That is, their analysis considered only the number of increments received. Our design for this metric results in a threshold penalty of at most 0.4 dB compared to the original PBRL ensemble of Chen et al. [CVDW15] for rate-compatibility in the usual sense, which was designed to have the best threshold at each rate.

The second metric sacrifices identical performance for every increment ordering to obtain RC codes with the best possible throughput. Here the focus is on enabling early successful decoding by minimizing the average protograph threshold over all orderings at each rate. This led to the best throughput in simulations where increments are provided to the decoder in a random order, as would occur with inter-frame coding.

While inter-frame coding is a motivation to consider rate-compatibility for any increment ordering, there are other scenarios where UIO codes may be deployed. A receiver may receive the incremental packets of an RC code from different relays where the relays might be susceptible to complete failure. In this case, a UIO code can guarantee an FER if a certain number of incremental packets are received, irrespective of what the incremental packets are from the point of view of the original RC code.

3.1.1 Design choices and remarks

We focus on design rates $8/i, i \in [10, 16]$ in this chapter. We design binary codes for binary-input additive white Gaussian noise channel (BI-AWGNC), with protograph iterative decoding thresholds computed using *reciprocal channel approximation (RCA)*. For a review of RCA as a one-dimensional approximation to density evolution of Richardson et al. [RSU01], see Chen et al. [CVDW15] and Divsalar et al. [DDJA09]. We focus on long block-lengths ($k = 16384$ information bits) to obtain codes that operate fairly close to their ensemble thresholds. We assume that the incremental bits are delivered in chunks corresponding to each protograph variable node. The design heuristic provided in this chapter is applicable to any memoryless symmetric channel, and the derived codes may be based also on non-binary Galois fields.

Remark 7. *A PBRL ensemble is completely specified by* P_{HRC}*, and* P_{IRC} *or a combination of its rows (See Chapter 1). It is implicitly assumed that the overall protograph is formed by appropriately including the necessary* 0 *and* I *matrices. Also, the degree-1 variable nodes in* P_{IRC} *do not depend upon each other. This facilitates the design of UIO-PBRL codes for the first metric.*

Remark 8. *The threshold values computed in this chapter are the result of at least 1000 iterations of the RCA method (See [CVDW15]). Codes simulated are quasi-cyclic and have a first-step lifting factor of 4 and a second-step lifting factor of 512. First-step lifting used Hu et al.'s progressive edge-growth (PEG) algorithm [HEA05], and second-step lifting used the circulant-PEG (C-PEG) algorithm. The ACE algorithm of Tian et al. [TJVW03] was also used in both steps. Simulation results shown were obtained using a maximum of 200 iterations of full-precision, flooding, LLR-domain belief propagation. At least 100 errors were collected for each FER point in any simulated* E_b/N_0 *vs. FER graph.*

The chapter is organized as follows: Section 3.2 presents ensembles for UIO-PBRL codes designed for the two metrics and shows simulation results of the new UIO-PBRL codes satisfying the requirements of the metrics. Section 3.3 concludes the chapter.

3.2 PBRL Ensembles for UIO-RC Codes

3.2.1 Long block-length PBRL ensemble of Chen et al. in [CVDW15]

Let us first recall how a PBRL ensemble for rate-compatibility in the usual sense is designed. First, we select an P_{HRC} with a degree distribution that has a good threshold and permits a low error floor. Then, each row of P_{IRC} is chosen by keeping all previous rows fixed and selecting edges for that row to obtain the best threshold possible while meeting constraints designed to preserve good error floor performance. The specific constraints and the complexity of the RCA algorithm dictate the overall complexity of this search. Note that, although this is a greedy search, one can obtain excellent thresholds and FER performance at all rates as demonstrated by Chen et al. in [CVDW15].

Let us first see how an original PBRL code from [CVDW15] behaves under different orderings of its increments. Consider the PBRL ensemble of [CVDW15] that was designed for long block-lengths with the best threshold at each rate. P_{HRC} and P_{IRC} ((13) and (14) in [CVDW15]), up to rate 8/16, are

$$\begin{aligned} P_{\mathsf{HRC}} &= \begin{bmatrix} 3&2&1&1&1&1&1&1&0&0&1 \\ 1&1&2&2&2&2&2&2&2&1&1 \\ 2&0&0&0&0&0&0&0&1&2&0 \end{bmatrix}, \\ P_{\mathsf{IRC}} &= \begin{bmatrix} 2&1&2&1&0&0&0&0&1&0&0 \\ 1&2&1&0&1&0&0&0&0&1&0 \\ 2&0&0&2&0&1&0&0&0&0&0 \\ 1&1&0&0&0&0&1&0&0&2&0 \\ 2&2&0&0&0&0&0&1&0&0&0 \\ 1&0&0&0&0&2&0&0&0&2&0 \end{bmatrix}. \end{aligned} \tag{3.1}$$

The design heuristics to obtain good thresholds for all code rates through the search in [CVDW15] led to the puncturing of the first protograph variable node in (3.1). We adopt the same principle in this chapter; our ensembles also have the first variable node punctured.

For the PBRL ensemble of (3.1), Table 3.1 shows the statistics of thresholds at each rate obtained by including various combinations of rows (corresponding to different orderings of the increments) of P_{IRC} with P_{HRC}. The number of ensembles at rate $8/(10+i), 0 \leq i \leq 6$ is $\binom{6}{i}$. From the table, we see a range of thresholds for all but the highest and lowest rates,

Table 3.1: Statistics of RCA Thresholds ($\mathrm{E}_b/\mathrm{N}_0$) over BI-AWGNC Considering P_{HRC} and Various Combinations of Rows of P_{IRC} in (3.1) for Rates $8/i, 10 \leq i \leq 16$. "Sh" Is the Shannon Limit.

Rate	Sh (dB)	Min.	Max.	Ave.	Std. Dev.	Max-Min
8/10	2.04	2.179	2.179	2.179	0	0
8/11	1.459	1.579	2.044	1.809	0.16	0.465
8/12	1.059	1.199	1.897	1.49	0.195	0.698
8/13	0.762	0.897	1.528	1.172	0.174	0.631
8/14	0.53	0.662	1.153	0.86	0.135	0.491
8/15	0.342	0.462	0.668	0.568	0.083	0.206
8/16	0.187	0.308	0.308	0.308	0	0

with the maximum gap being 0.698 dB. The ensemble in (3.1) shows considerable variation in threshold for different orderings of the increments. Simulation results of the code obtained from this ensemble in [CVDW15] at rate 8/11, shown in Fig. 3.1, illustrate how the threshold variation manifests itself in FER performance.

3.2.2 Design metric 1: UIO-PBRL codes that have, at each rate, identical FER performance for every ordering

This subsection seeks codes that have, at each rate, identical FER performance for every increment ordering. This metric requires that the thresholds of various ensembles at each rate formed using the different possible sets of rows of P_{IRC} be as close as possible to each other. This means that the rows of P_{IRC} need to be chosen such that the threshold gap at each rate (as in Table 3.1) is as small as possible. Also, to be considered a good ensemble, the minimum threshold at each rate should be as small as possible. A small threshold gap at each rate mandates some symmetry in the ensemble, and the minimum threshold at each rate depends upon the degree distribution.

We restrict the maximum non-zero value in P_{IRC} to be equal to 2 in order to reduce the search complexity. Given an P_{HRC}, initial attempts to produce an P_{IRC} included making

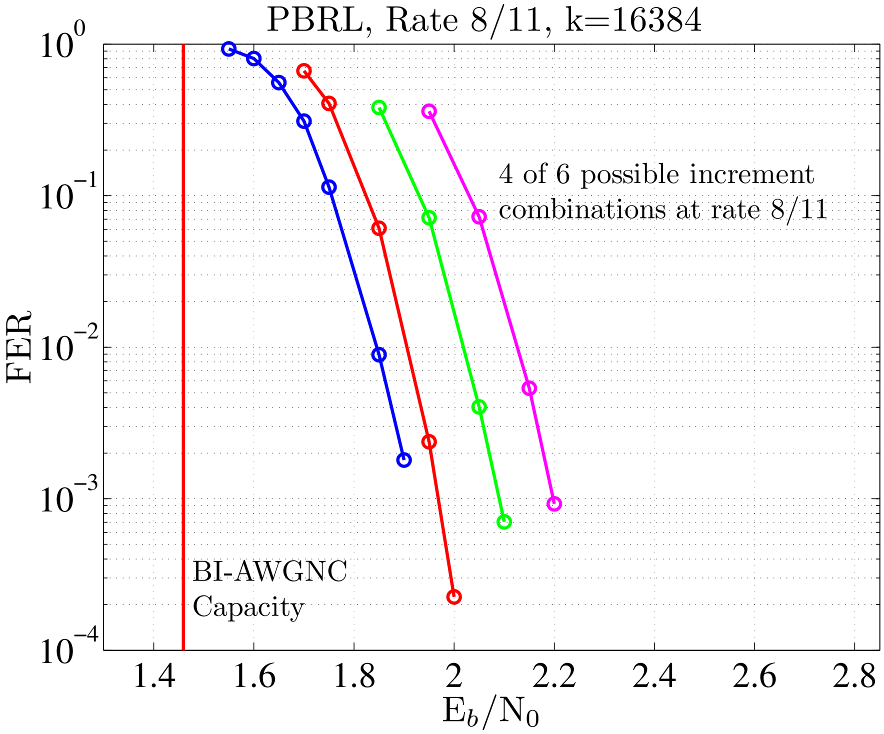

Figure 3.1: Dependence of a PBRL code on the ordering of its increments

sure that each row of P_{IRC} has the same weight and that all rows have the same number of different types of non-zero values. We restricted the first column of P_{IRC} to either have all ones or all twos. The all-twos designs led to codes with poor FER performance. Our initial exhaustive search led to the following:

$$\begin{aligned} P_{\mathsf{HRC}} &= \begin{bmatrix} 3&1&1&1&1&1&1&2&2&2&0 \\ 1&1&2&2&2&2&2&1&1&1&0 \\ 2&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}, \\ P_{\mathsf{IRC}} &= \begin{bmatrix} 1&0&0&0&0&1&1&1&1&1&0 \\ 1&0&0&0&1&0&1&1&1&1&0 \\ 1&0&0&0&1&1&0&1&1&1&0 \\ 1&0&0&0&1&1&1&0&1&1&0 \\ 1&0&0&0&1&1&1&1&0&1&0 \\ 1&0&0&0&1&1&1&1&1&0&0 \end{bmatrix}. \end{aligned} \tag{3.2}$$

Table 3.2: Statistics of RCA Thresholds (E_b/N_0) for Ensemble in (3.2)

Rate	Sh (dB)	Min.	Max.	Ave.	Std. Dev.	Max-Min
8/10	2.04	2.393	2.393	2.393	0	0
8/11	1.459	1.892	1.896	1.894	0.0022	0.004
8/12	1.059	1.494	1.502	1.498	0.0026	0.008
8/13	0.762	1.148	1.158	1.1539	0.0026	0.01
8/14	0.53	0.856	0.862	0.8596	0.002	0.006
8/15	0.342	0.640	0.642	0.641	0.0011	0.002
8/16	0.187	0.506	0.506	0.506	0	0

Notice that every row in P_{IRC} is "equivalent" in that within P_{IRC} two rows can be exchanged and the matrix is still the same with a simple column exchange thereafter. Table 3.2 shows the statistics of the ensemble in (3.2). The maximum of all threshold gaps is 0.01 dB compared with 0.698 dB in Table 3.1.

Now, keeping P_{IRC} the same as in (3.2), we present an ensemble with zero gap between the thresholds at each rate:

$$\begin{aligned} P_{\mathsf{HRC}} &= \begin{bmatrix} 3\;1\;1\;1\;1\;1\;1\;1\;1\;1\;0 \\ 1\;1\;2\;2\;2\;2\;2\;2\;2\;2\;0 \\ 2\;0\;0\;0\;0\;0\;0\;0\;0\;0\;1 \end{bmatrix}, \\ P_{\mathsf{IRC}} &= \begin{bmatrix} 1\;0\;0\;0\;0\;1\;1\;1\;1\;1\;0 \\ 1\;0\;0\;0\;1\;0\;1\;1\;1\;1\;0 \\ 1\;0\;0\;0\;1\;1\;0\;1\;1\;1\;0 \\ 1\;0\;0\;0\;1\;1\;1\;0\;1\;1\;0 \\ 1\;0\;0\;0\;1\;1\;1\;1\;0\;1\;0 \\ 1\;0\;0\;0\;1\;1\;1\;1\;1\;0\;0 \end{bmatrix}. \end{aligned} \tag{3.3}$$

By the virtue of apparent symmetry in the ensemble in (3.3) (taking into consideration P_{HRC} and P_{IRC} together), it is clear that the threshold gap, Max.−Min., is zero at each rate.

Table 3.3: RCA Thresholds (E_b/N_0, decibel) for Ensemble in (3.3)

Rate	8/10	8/11	8/12	8/13	8/14	8/15	8/16
Sh	2.04	1.459	1.059	0.762	0.53	0.342	0.187
Thr.	2.462	1.934	1.518	1.156	0.842	0.606	0.474

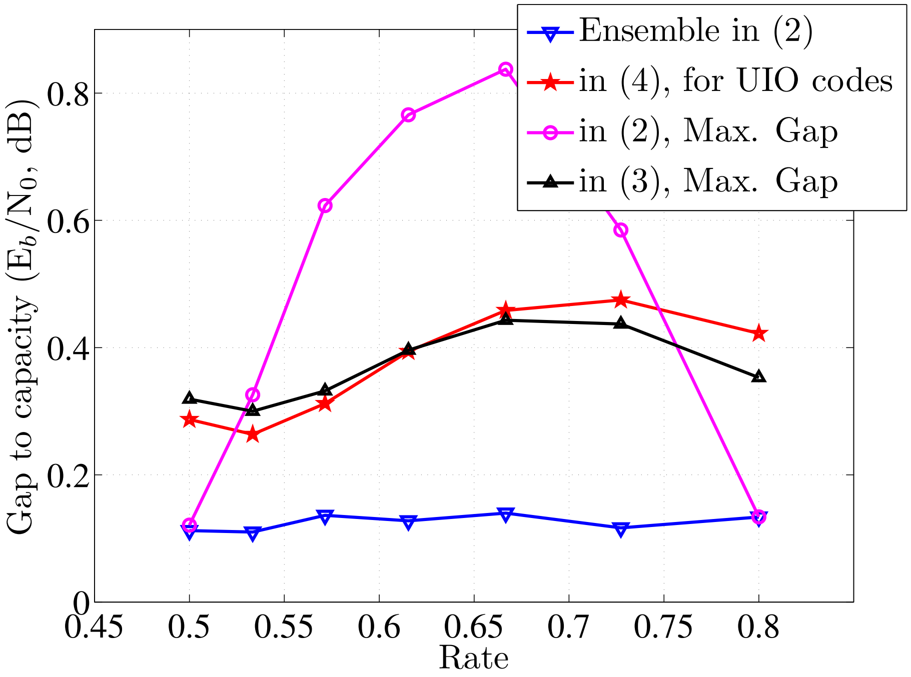

Figure 3.2: Gap to BI-AWGNC capacity – Ensembles in (3.1), (3.2), (3.3)

Table 3.3 presents the computed threshold at each rate.

Fig. 3.2 plots the gap to capacity of the original PBRL ensemble for long block-lengths in (3.1) and the ensemble in (3.3). Also plotted are the gaps to capacity of the worst-case threshold (Max.) for the PBRL ensembles in (3.1) and (3.2)

We simulated a code from the ensemble in (3.3), and the results are shown in Figs. 3.3 and 3.4. A zero threshold gap at each rate certainly seems to be a predictor of *exactly* the same performance for every increment ordering. We observed the same at every rate, and the results at other rates are not included for brevity. Note that the length of the code drawn from the ensemble may play a role in the actual performance due to local graph effects. We intend to investigate if similar results hold for much shorter block-lengths.

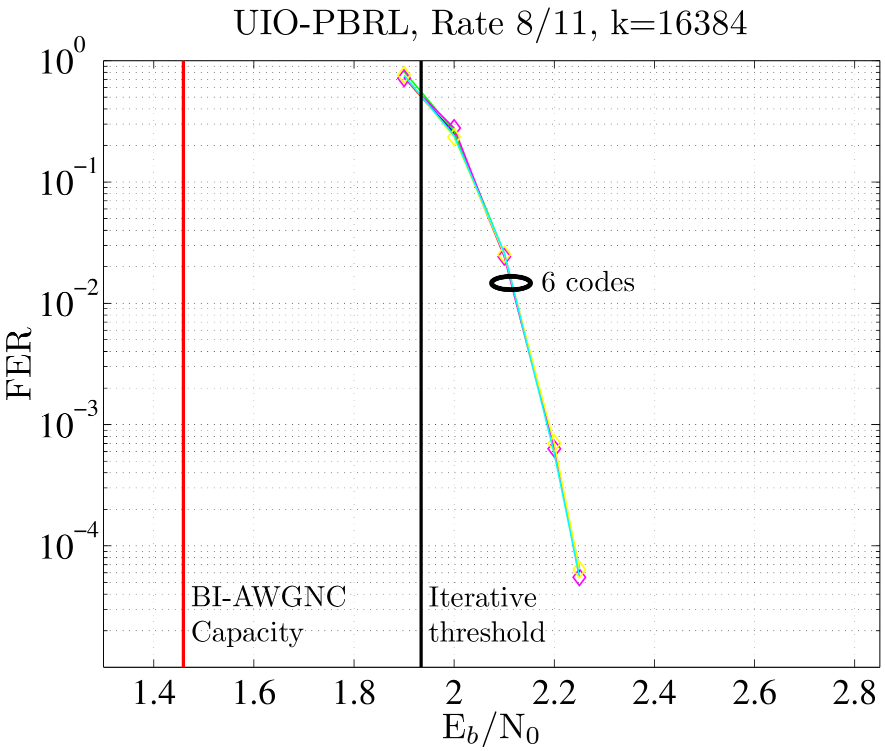

Figure 3.3: Simulations of a UIO-PBRL code that show that the code has exactly the same performance irrespective of the ordering of its increments

3.2.3 Design metric 2: UIO-PBRL codes with the best throughput over all increment orderings for inter-frame coding

The metric of minimizing threshold gap does not necessarily maximize the throughput in inter-frame coding. Our second design metric seeks codes that attempt to maximize throughput in inter-frame coding by decoding as early as possible. The simulations in this subsection are carried out according to how an RC code in an inter-frame code operates. The decoder starts decoding at the highest rate. If it is unsuccessful, a randomly chosen increment becomes available to the decoder. The process is repeated until the decoder decodes successfully or fails at the lowest rate. The throughput, in simulations, is the ratio of total number of information bits delivered successfully to the total number of codeword bits sent over the channel.

First, we compare the throughput of the original PBRL code for 16384 information bits

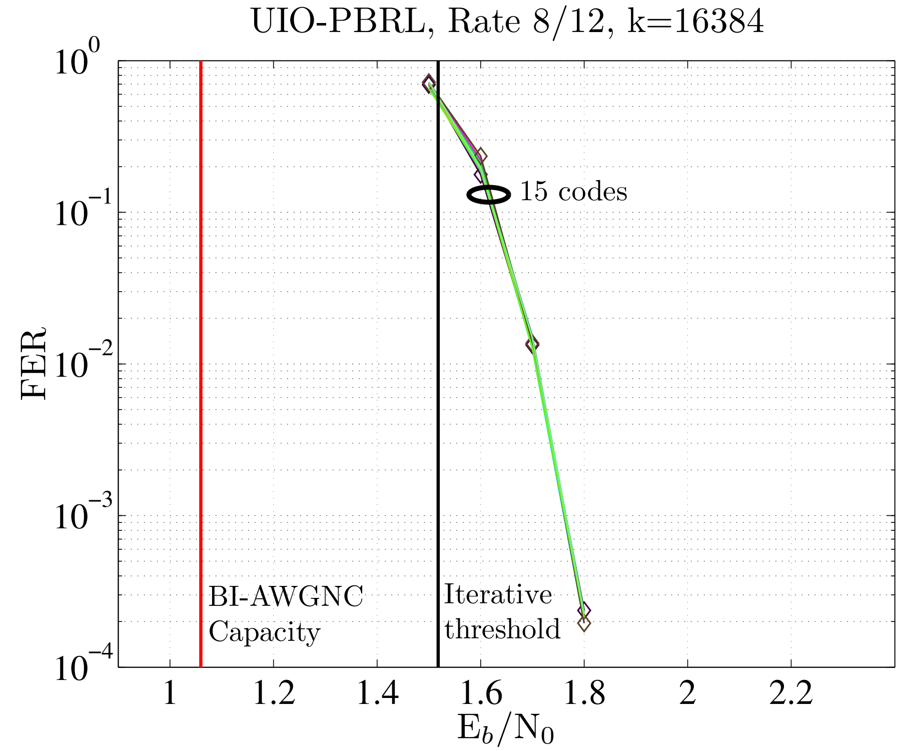

Figure 3.4: Strictly identical performance at rate 8/12 for all 15 orderings

(ensemble in (3.1)) and a UIO-PBRL code designed according to the first metric with the same P_{HRC} as in (3.1). The results, in Fig. 3.5, show that the original PBRL code has a higher throughput than the UIO-PBRL code at most channel SNRs. We do not provide the ensemble for this UIO-PBRL code for brevity. It has a maximum threshold gap of 0.02 dB and a gap of 0 dB at rates 8/10, 8/11, 8/15, and 8/16. As each increment ordering is equiprobable, one metric that affects throughput is the average threshold at each rate. We found that the average threshold at each rate for the ensemble of the UIO-PBRL code is greater than or equal to the average threshold of the ensemble of the original PBRL code (Table 3.1).

Based on this, we modified the PBRL design process to obtain an ensemble with low average thresholds. For this design, the P_{HRC} is the same as in (3.1). The original PBRL search designs P_{IRC} one row at a time to avoid exponentially high complexity. We retained this general procedure but widened our search space as follows: As each row is added, we

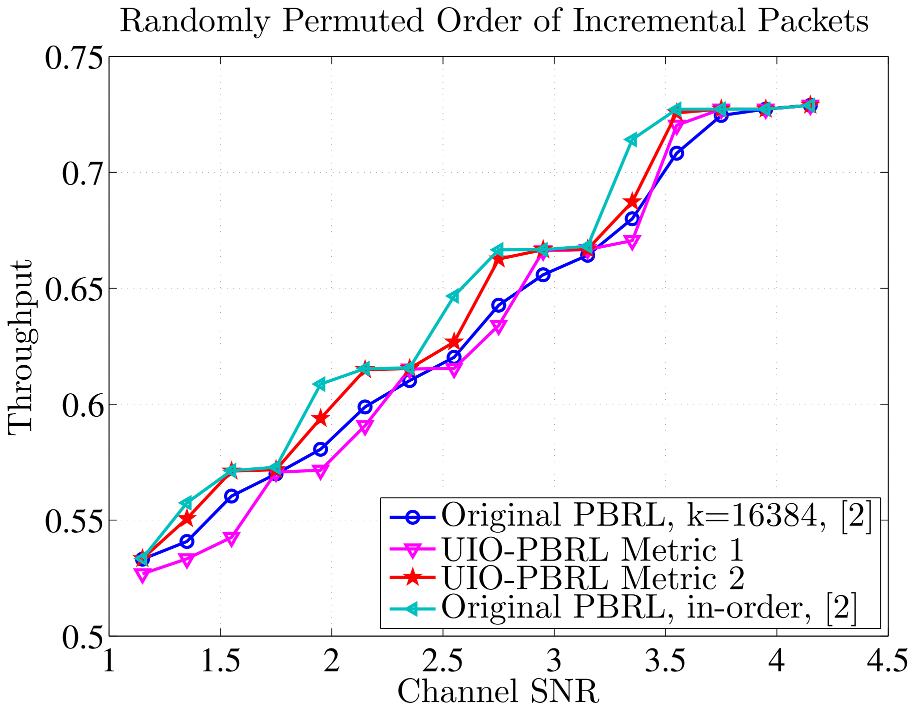

Figure 3.5: Throughput comparison

considered the 3 best ensembles in terms of the threshold at that rate. That is, starting with the P_{HRC} at rate 8/10, we obtained 3 ensembles at rate 8/11, 9 at rate 8/12, and so on. We discarded isomorphic ensembles in obtaining the 3 best ensembles at each rate for a given matrix from the previous rate. From the resulting 729 ensembles, we obtained the ensemble in (3.4) with the best average threshold for all rates except 8/16 (Table 3.4).

$$P_{\mathsf{IRC}} = \begin{bmatrix} 2 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.4}$$

Fig. 3.5 shows that this approach has led to the best throughput at each channel SNR. Note that the two design metrics seem to be conflicting with each other. Also shown here is

Table 3.4: Average RCA Threshold (E_b/N_0, db) at Each Rate for Ensemble in (3.4) Compared Against Original PBRL Ensemble in [CVDW15]

Rate	8/11	8/12	8/13	8/14	8/15	8/16
Avg. Thr., (3.4)	1.717	1.328	0.998	0.727	0.515	0.394
cf. Table 3.1, (3.1)	1.809	1.49	1.172	0.86	0.568	0.308

the throughput of the original PBRL code [CVDW15] when its incremental packets arrive in order. Since the ensemble of this code has the best possible threshold at each rate, the throughput of this code when its increments are appended in order is the maximum that is possible for a PBRL code. Our code for metric 2 comes close to the best throughput at many channel SNR values.

3.3 Concluding Remarks

We designed rate-compatible codes with universally good performance for any increment ordering. The chapter used two design metrics to obtain such codes, which are called UIO codes. One metric requires, at each rate, the same performance for all increment orderings. The other metric sacrifices identical performance for every ordering to seek codes that have the best average threshold at each rate.

Acknowledgment

The work that resulted in this chapter was supported in part by NSF grants CCF-1162501 and CCF-1161822 (JPL Task Plan 82-17473). Research was carried out in part at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with NASA.

CHAPTER 4

On the Girth of (3,L) Quasi-Cyclic LDPC Codes based on Complete Protographs

This chapter[1] turns from PBRL codes away to general protograph LDPC codes. Consider the problem of constructing $(3, L)$ QC-LDPC codes from complete protographs. A complete protograph is a small bipartite graph with two disjoint vertex sets such that every vertex in the variable-node set is connected to every vertex in the check-node set by a unique edge. This chapter analyzes the required lifting factor for achieving girths of six or eight in the resulting quasi-cyclic codes with constraints on lifting. The required lifting factors provide lower bounds on the block-length of such codes.

4.1 Introduction and Background

A protograph [Tho03] defines the family of codes that can be obtained from it by lifting and many properties of the codes in the family depend on the graphical structure of the chosen protograph. In this chapter, we consider the case where the protograph is a *simple* (has no loops or multiple links between two vertices) and *complete* (every vertex in the variable-node set is connected to every vertex in the check-node set) bipartite graph. QC-LDPC codes obtained from simple and complete protographs are called *conventional* QC-LDPC codes in [KB13]. The authors of [KB13] consider simple QC-LDPC codes in general, including the

[1] Almost all of this chapter was presented at the 2015 IEEE International Symposium on Information Theory (ISIT) [RDW15a]. An open-source version appeared in [RDW15b].

subset which are conventional.

The performance of LDPC codes is dictated, to a certain extent, by the girth of the codes. Also, in the regime of short-to-moderate block-lengths, the minimum distance of an LDPC code affects its performance in the error-floor region if the variable-node degrees are small [RDVW14]. In this regard, the minimum distance of a protograph QC-LDPC code and its girth are interrelated as suggested by the work in [SV12]. The works in [KB13, Fos04] derive lower bounds on the required lifting factors (and thus block-lengths) for obtaining various girths for QC codes constructed from protographs and provide the foundations for this chapter. Works including [Fos04, KB13, KCY13, WYD08, HNK$^+$06] have focused on obtaining these bounds because they are of practical importance and have demonstrated code construction techniques to obtain codes with as high a girth as possible.

We focus on the case of $(3, L)$ protograph QC-LDPC codes. These are regular codes that perform well over many rates. The chapter is organized as follows: Section 4.2 introduces notation. Section 4.3 completely characterizes the lifting requirements to construct a $(3, L)$ code with girth 6 when the lifting factor is equal to L and gives an explicit construction that achieves a girth of 6 for any possible value of L. Section 4.4 derives a bound (under a constrained setting) on the lifting factor required to obtain a girth of at least 8. This bound improves on the bounds in [KB13, Fos04]. Section 4.5 concludes the chapter.

4.2 Definitions and Notation

The protomatrices considered in this chapter have the form

$$H_{\text{protomatrix}} = \begin{bmatrix} 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & \cdots \end{bmatrix}_{3\times L} . \tag{4.1}$$

At places, the terms protograph and protomatrix are used interchangeably. Associated with any protomatrix, the process of *lifting* to obtain a QC code is the replacement of every non-zero entry z in the protomatrix by a sum of z circulant permutation matrices (CPMs) of size $N \times N$ with distinct support and every 0 in the protomatrix by an $N \times N$ matrix of all zeros. If the protomatrix is of size $J \times L$ then lifting yields a parity-check matrix H of size $JN \times LN$. Because our protomatrices are simple and complete, lifting replaces every entry in the protomatrix with an $N \times N$ CPM.

Definition 3 (Permutation-shift matrix [KB13])**.** *The permutation-shift matrix P of a QC-LDPC code constructed from a $J \times L$ protomatrix with entries at most equal to 1 is the $J \times L$ matrix of permutation shift indices that are chosen for the non-zero entries of the protomatrix during the process of lifting. With the lifting factor being N, an element $0 \leq x \leq N-1$ in P corresponds to a CPM in the parity-check matrix H obtained via x circular shifts of the rows of the identity matrix of size $N \times N$. The orientation (left or right) of the permutation shifts is unspecified in this chapter without loss of generality (WLOG).*

The cyclic group of integers modulo N, $\{0, 1, \ldots, N-1\}$, is denoted $\mathbb{Z}/N$. This is the set of first N non-negative integers with addition modulo-N as the associated binary operation, represented by $x_i + x_j$. Similarly, $x_i - x_j = x_i + (-x_j)$ represents adding the inverse of x_j to x_i. The *order* of a group is its cardinality. A *permutation* π is a bijective map of a finite set of elements onto itself.

Definition 4. *A permutation π of $\mathbb{Z}/N$ is said to have a* fixed point *if $\pi(i) = i$ for any $i = 0, 1, \ldots, N-1$.*

4.3 On the Minimum Lifting Factor for Girth Greater Than or Equal To 6

We consider the special case of this problem with the constraint that the lifting factor N satisfies $N = L$. This is the least value of N for which one can possibly obtain a girth of $g >$

4 [Fos04, KB13]. By looking at this special case we arrive at a combinatorial interpretation to the problem of obtaining codes with girth at least 6 from complete protomatrices of size $J \times L$. Since $N = L$, we may use N and L interchangeably.

Works including [Fos04] have constructed codes via computer searches to show empirically the existence of codes with girth $g \geq 6$ for some odd values of $N = L$ (including analytical constructions for all primes; see [Fan00] also). We show analytically that for all odd values of $N = L$, there exist $(3, L)$ codes with girth $g \geq 6$. [HNK^{+}06] has established this result and our contribution is a proof via combinatorial structures called *complete mappings* [Pai47]. We provide an algebraic construction that produces codes with girth $g = 6$ for any odd $N = L$. This construction includes, as a special case, the array-code based proof of [HNK^{+}06] for the $(3, L)$ case.

Lemma 5 ([Fos04])**.** *With the lifting factor being N, in any QC-LDPC code with a protomatrix with no entry larger than 1, a cycle of length ℓ (ℓ even) in the Tanner graph of the code can be equivalently described by a sequence of edges $(e_1, e_2, \ldots, e_\ell)$ in the protograph whose corresponding permutation shifts in P that are given as $x_1, x_2, \ldots, x_\ell$ satisfy*

$$\sum_{i=1}^{\ell}(-1)^{i+1}x_i = 0 \mod N, \tag{4.2}$$

where $e_i \neq e_{i+1}$ for all $i \in \{1, 2, \ldots, \ell - 1\}$ and $e_1 \neq e_\ell$. Consecutive pairs of consecutive edges $\{e_i, e_{i+1}\}$ for all $i \in \{1, 2, \ldots, \ell - 1\}$ and $\{e_\ell, e_1\}$ alternatingly lie in the same row or same column of the protomatrix.

The elements of P are assumed to be in $\mathbb{Z}/N$ and thus “mod N" may not be mentioned at most places that involve operations with elements from P.

Lemma 6 (Extension of [Fos04], Theorem 2.2)**.** *With a lifting factor of $N = L$, any permutation-shift matrix P that could lead to $g > 4$ for a $(3, L)$ code with a complete pro-*

tomatrix may be written WLOG as

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & N-1 \\ \pi(0)=0 & \pi(1) & \pi(2) & \cdots & \pi(N-1) \end{bmatrix}, \tag{4.3}$$

where π has only one fixed point at $\pi(0) = 0$.

Proof. Irrespective of the $3L$ indices that are chosen for P, one can always apply circular shifts to the row blocks and the column blocks of H (after lifting) to obtain an isomorphic graph for which the first row and column have all-zero indices in P, as observed in [Fos04]. For girth $g > 4$, [Fos04] shows that no non-zero element can repeat in the same row or the same column. Thus the non-zero entries in each of rows 2 and 3 are all unique within the respective rows and the ordering of row 2 in (4.3) can be obtained WLOG by rearranging the columns once we have 0's in row 1 and column 1. To ensure that no column repeats a nonzero value, the permutation cannot have any fixed point except $\pi(0) = 0$. □

The preceding lemma implies that, WLOG, only $L - 1$ non-zero permutation shifts need to be specified and these belong to the third row. As an example where repetition in the same column prevents $g > 4$, the case of $L = 2$ leads to $g = 4$ as there is only one non-zero element in $\mathbb{Z}/2$. The probability that a permutation of a finite number of elements $(N - 1)$ has no fixed points asymptotically, as $N \to \infty$, equals $\frac{1}{e}$ [MN08]. If we search randomly for permutations of $N - 1$ non-zero elements to achieve a girth of $g > 4$, then the number of permutations to be considered when constructing a code for large values of $N - 1$ is very high but only approximately 36.8% of them will pass the preliminary test of not having a fixed point.

Definition 5 (Complete mapping [Pai47, Slo])**.** *A complete mapping of the cyclic group $(\mathbb{Z}/N, +)$ is a permutation π which satisfies $\pi(0) = 0$ and that $(0, \pi(1)-1, \pi(2)-2, \ldots, \pi(N-1) - (N - 1))$ is also a valid permutation.*

Theorem 8. *With a lifting factor of $N = L$, the parity-check matrix H of a code with a complete protomatrix of size $3 \times L$ has a girth $g > 4$ if and only if the permutation π of $\mathbb{Z}/N$ that specifies the third row of P in (4.3) is a complete mapping.*

Proof. Consider any two columns of the shift matrix of (4.3) and form a 2×2 sub-matrix of rows 2 and 3 out of the chosen columns as

$$\begin{bmatrix} x_i & x_j \\ x_k & x_\ell \end{bmatrix}, x_k = \pi(x_i), x_\ell = \pi(x_j).$$

From the general condition of (4.2) in Lemma 5, x_i, x_j, x_k, x_ℓ lead to cycle(s) of length four if and only if (iff)

$$x_i - x_k + x_\ell - x_j = 0. \tag{4.4}$$

Rewriting the above, the girth is greater than 4 iff

$$(x_\ell - x_j) - (x_k - x_i) \neq 0, \tag{4.5}$$

which means that $x_\ell - x_j \neq x_k - x_i$ should be satisfied for any x_i, x_j, x_k, x_ℓ as considered above. This is possible iff

$$(\pi(\text{row } 2) - \text{row } 2) \tag{4.6}$$

describes a permutation (i.e. the sequence contains each distinct element in the group exactly once), which occurs iff row 3 is a complete mapping. □

Theorem 9. *There exists a $(3, L)$ quasi-cyclic LDPC code with a complete protograph lifted by a factor $N = L$ satisfying girth $g > 4$ iff L is odd.*

Proof. From [Pai47], there exists a complete mapping of a finite abelian group of order N iff the group does not possess exactly one element of order 2. When N is even, this condition is violated as one can verify that $\frac{N}{2}$ is the only order-2 element in the finite abelian group $\mathbb{Z}/N$. On the contrary, in finite groups $\mathbb{Z}/N$ of odd orders there exists no element of order 2, according to Lagrange's theorem on the order of elements in a finite group. This argument in conjunction with Theorem 8 completes this proof. □

The number of complete mappings of $\mathbb{Z}/N$ is documented in [Slo]. The first few terms of this sequence as a function of N, from $N = 1, 3, 5, \ldots$, are 1, 1, 3, 19, 225, 3441, 79259, 2424195, 94471089, 4613520889. For odd $N = L$ all the complete mappings that yield codes with girth $g \geq 6$ lead to $g = 6$ since girth $g \geq 8$ requires a higher lifting factor (see Section 4.4). For odd $N = L$, random search might identify a complete mapping and hence a $g = 6$ code, but the probability of any randomly selected mapping being complete decreases quickly with increasing L. For instance, when $L = 15$ corresponding to a design rate $R = \frac{L-3}{L} = 0.8$ this probability is $\frac{2424195}{14!} = 0.000028$ and when $L = 17$ and $R = 0.8235$ this probability is 0.000004 and so on. In the following we present a family of complete mappings and thus a family of codes for any odd $N = L, L \geq 3$ that have $g = 6$.

Corollary 2 (Product construction)**.** *Consider the following mapping for row 3 in (4.3) with* $h \in \{2, 3, \ldots, N-1\}$*:*

$$\pi_p(i) = hi \mod N, 0 \leq i \leq N-1, \tag{4.7}$$

where $hi \mod N$ *is multiplication modulo-*N *of integers* h *and* i*. For* $N = L$ *odd and* $N \geq 3$*, if* h *and* $h-1$ *are each coprime with* N*, then* π_p *is a complete mapping of* $\mathbb{Z}/N$ *and thus leads to a* $(3, L)$ *code with girth 6.*

Proof. Note that since h is chosen to be coprime with respect to N, $(hi \mod N : 0 \leq i \leq N-1)$ is a valid permutation of $\mathbb{Z}/N$. This is because $hi - hj = h(i-j) \neq 0 \mod N, \forall i \neq j$ as h is not a factor of N. We need to further show that (4.4) from Theorem 8 has no solution.

Writing (4.5), which is obtained from (4.4), for this permutation:

$$\begin{aligned}
&(x_\ell - x_j) - (x_k - x_i) \neq 0 \\
\iff &h(x_j - x_i) - (x_j - x_i) \neq 0, \text{ as } x_k = hx_i, x_\ell = hx_j \\
\iff &(h-1)(x_j - x_i) \neq 0,
\end{aligned}$$

which is satisfied for this permutation for all $x_j \neq x_i$ since $h - 1 \geq 1$ is chosen to be coprime with respect to N. □

There exists such an h for every odd $N \geq 3$. An example is $h = N - 1$ for which

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 2 & \cdots & N-2 & N-1 \\ 0 & N-1 & N-2 & \cdots & 2 & 1 \end{bmatrix}. \tag{4.8}$$

Also, $(3, L)$ array codes [Fan00, Fos04], for any odd $L \geq 3$ (not necessarily prime), are a special case of the preceding construction with $h = 2$ and thus have $g = 6$ [HNK$^+$06].

Corollary 3. *If $N = L$ is even then there exists a $(3, L)$ complete-protomatrix-based code with girth equal to 4 whose Tanner graph has exactly N cycles of length four.*

Proof. This follows from [Pai47], which proves that in case the order of a finite abelian group is even then there exists an "almost complete" mapping π of the group such that the sequence $(0, \pi(1) - 1, \pi(2) - 2, \ldots, \pi(N-1) - (N-1))$ has exactly $N - 1$ distinct elements. Thus, one element appears twice. This implies that there exists a mapping for the third row such that only one 2×2 block from the second and the third rows leads to N length-4 cycles. □

As shown in [HNK$^+$06], it can also be observed that if L is even then there exists a $(3, L)$ complete-protomatrix-based code with a girth of 6 if the lifting factor is $N = L + 1$.

One can generalize the discussion so far to see that for the (J, L) case there could be a code with $g > 4$ when the lifting factor is $N = L$ only if there exist $J - 2$ distinct complete mappings of $\mathbb{Z}/N$. This condition is necessary but not sufficient because the rows produced by the $J-2$ complete mappings also have to satisfy the following condition: Every pair of the $\binom{J-2}{2}$ rows indexed by $\{\{i,j\} : 3 \leq i < j \leq J\}$ are such that row j is a complete mapping of row i.

Consider the computer-search based Table 4.1 of [Fos04] (reproduced below). When $N = L = 9$, the computer search could not find a $(J, 9)$ code with girth $g = 6$ when $J \geq 4$. Using the previous paragraph, we can confirm that such a code does not exist. There are 225 complete mappings of $\mathbb{Z}/9$. We can corroborate the result in this table since not even one pair out of $\binom{225}{2}$ pairs of complete mappings can satisfy the requirement that one row in the pair is a complete mapping of the other.

Table 4.1: Smallest Value of N for which a (J, L) Code with Girth $g \geq 6$ was Found in [Fos04] Using Computer Search

L J	4	5	6	7	8	9	10	11	12
3	5	5	7	7	9	9	11	11	13
4	–	5	7	7	9	**10**	11	11	13
5	–	–	7	7	9	**10**	11	11	13

4.4 Towards a Tighter Bound on the Required Lifting Factor for Girth $\geq$ 8 while $L \geq 4$

Assuming $L \geq 4$, it is known that the lifting factor N has to satisfy $N > 2(L-1)$ to obtain a girth of $g \geq 8$ for our $(3, L)$ codes [Fos04]. In this section, we derive an improved bound on this required lifting factor under a constraint by using an additive combinatorics formulation of the problem. It is conjectured, for future investigation, that the bound holds without this imposed constraint.

The following lemma states the necessary and sufficient conditions of [Fos04] for the permutation-shift matrix P of a complete-protomatrix-based $(3, L)$ code to achieve $g \geq 8$.

Lemma 7. *For $L \geq 4$, let $L' = L - 1$ and the lifting factor be N. The permutation-shift matrix*

$$P = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & x_1 & x_2 & \dots & x_{L'} \\ 0 & x_{L'+1} & x_{L'+2} & \dots & x_{2L'} \end{bmatrix} \tag{4.9}$$

leads to a girth of $g \geq 8$ iff all the following conditions hold: With $i, j \in \{1, 2, \dots, 2L'\}$,

1. *$x_i \neq x_j$ for all $i \neq j$ and $x_i \neq 0$ for all i.*

Fixing $i \geq L' + 1$ and $j = i - L'$ (so that x_i and x_j are in the same column of P, with x_i in the third row):

2. *$x_i - x_j \neq -x_k$, where $k \in \{1, 2, \dots, L'\} \setminus \{j\}$,*

3. *$x_i - x_j \neq x_k$, where $k \in \{L' + 1, L' + 2, \dots, 2L'\} \setminus \{i\}$,*

4. *$x_i - x_j \neq x_k - x_\ell$, where $k \in \{L' + 1, L' + 2, \dots, 2L'\} \setminus \{i\}, \ell \in \{1, 2, \dots, L'\} \setminus \{j\}, k \neq \ell + L'$,*

5. *$x_i - x_j \neq x_k - x_\ell$, where $k \in \{L' + 1, L' + 2, \dots, 2L'\} \setminus \{i\}, k = \ell + L'$.*

Proof. Condition 1 is Theorem 2.4 of [Fos04], which yields the necessary condition $N > 2(L - 1) = 2L'$ for achieving $g \geq 8$. Conditions 2 and 3 apply (4.2) to the first column and any other two columns of the shift matrix in (4.9). Condition 4 similarly considers any three columns apart from the first (all-zeros) column. Condition 5 avoids length-4 cycles from rows 2 and 3 of P. □

Definition 6 (Girth-8 table)**.** *A* girth-8 table *(G_8 table) of a $(3, L)$ complete-protomatrix-based QC-LDPC code whose permutation-shift matrix is P, using the notation of Lemma 7, is a table of $L' \times L'$ differences:*

A valid G_8 *table is one which leads to a girth of $g \geq 8$.*

+\-	x_1	x_2	$\dots$	$x_{L'}$
$x_{L'+1}$	$d_1 = x_{L'+1} - x_1$	$x_{L'+1} - x_2$	$\dots$	$x_{L'+1} - x_{L'}$
$x_{L'+2}$	$x_{L'+2} - x_1$	d_2	$\dots$	$x_{L'+2} - x_{L'}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x_{2L'}$	$x_{2L'} - x_1$	$x_{2L'} - x_2$	$\dots$	$d_{L'}$

Lemma 8. *A G_8 table is valid iff*

1. *The set of row and column headers together has $2L'$ distinct non-zero elements,*
2. *The diagonal elements $d_1, d_2, \dots, d_{L'}$ are all different from the inverses of the column headers,*
3. *The diagonal elements are all different from the row headers,*
4. *None of the diagonal elements is equal to any of the off-diagonal elements of the table,*
5. *The diagonal elements are all distinct.*

Proof. These conditions are the equivalent conditions of Lemma 7 in the same order. Note that a valid G_8 table has no 0 anywhere in it. Conditions 4 and 5, which are mathematically the same albeit for the choice of elements involved but stated separately for clarity, according to Lemma 7, justify uniquely identifying the diagonal elements as $d_1, d_2, \dots, d_{L'}$. □

Theorem 10. *Let the L' rows of any valid G_8 table be considered as sets of L' elements each and denoted $A_1, A_2, \dots, A_{L'}$. If there exist two rows $i \neq j$ such that $|A_i \cap A_j| = 0$ or $|A_i \cap A_j| = L' - 1$ then such a valid G_8 table corresponds to a lifting factor of $N \geq 3L' - 1$.*

Proof. In general, $|A_i \cap A_j| \leq L' - 1, i \neq j$ since every row has a diagonal element that is distinct from the elements in the rest of the table. The proof, which is given in the rest of this section, applies conditions 1, 4 and 5 from Lemma 7.

The case where $\exists\ i \neq j : |A_i \cap A_j| = 0$ is considered first. If so, then $|A_i| + |A_j| = 2L'$ and the rest of the $L' - 2$ rows contribute at least one distinct element each as their diagonal elements have to be distinct and thus the number of distinct non-zero elements is at least $3L' - 2$ and $N \geq 3L' - 1$.

For the second case, assume WLOG that the rows i, j are the first two rows of the G_8 table, corresponding to A_1 and A_2, or the table can be rearranged accordingly (this corresponds to permuting the columns of P). Denote the L' distinct elements of A_1 (in order from left to right) as

$$d_1 = x_{L'+1} - x_1, f_1, f_2, \ldots, f_{L'-1}.$$

Any $A_i, i \neq 1$ can be derived from A_1 through an offset. For example, A_2 can be obtained from A_1 by adding $\Delta = x_{L'+2} - x_{L'+1}$ to $d_1, f_1, f_2, \ldots, f_{L'-1}$ in that order.

The supposition $|A_1 \cap A_2| = L' - 1$ implies that A_1, A_2 differ in only $d_1 \neq d_2$. Since d_1 does not repeat or “lead to" a new element (or else $|A_1 \cap A_2| < L' - 1$), while adding $\Delta \neq 0$ to it and since the only new element that is formed in this second row is $d_2 = f_1 + \Delta$, this means that $d_1 + \Delta = f_i$ for some $i \in \{1, 2, \ldots, L' - 1\}$. Also $\forall k \in \{2, 3, \ldots, L' - 1\}$ there exists a unique $\ell_k \in \{1, 2, \ldots, L' - 1\} \setminus \{k\}$ such that $f_k + \Delta = f_{\ell_k}, f_k + \Delta \neq d_1, f_k + \Delta \neq d_2$.

Definition 7 (Circular representation)**.** *We choose to represent the elements of $\mathbb{Z}/N$ as unique points on a circle in order from 0 through $N - 1$ in the anticlockwise direction such that $N - 1$ appears on the circle before crossing 0 when counting from 0 (as integers). With this representation, addition corresponds to moving along the anticlockwise direction.*

4.4.1 Case 1

Figs. 4.1 and 4.2 show one possible case for the structure of the elements in $A_1 \cup \{d_2\}$ with respect to the circular representation. The figures capture one possible case, with no wrap around and wrap around shown separately. In both the figures, the $L' + 1$ elements $d_1, \ldots, f_1, d_2$ form a single chain from d_1 to d_2, through L' intervals of $\Delta = x_{L'+2} - x_{L'+1}$ points each, as shown in Figs. 4.1 and 4.2. WLOG the chain begins at d_1 and progresses

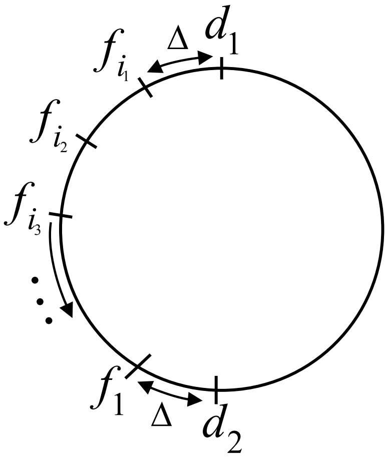

Figure 4.1: One possible arrangement of elements of row 1 for Theorem 10 – without wrap around

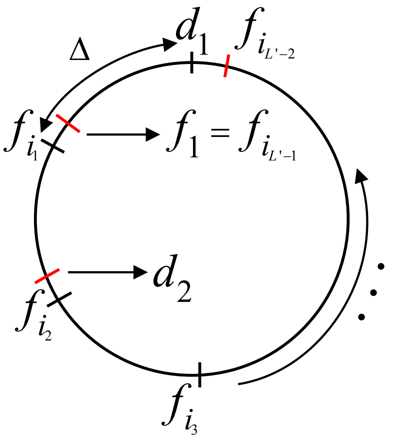

Figure 4.2: Another possible arrangement of elements of row 1 for Theorem 10 – with wrap around

anticlockwise or else the two rows can be exchanged to yield this.

Lemma 9. *For a valid G_8 table that falls in the case illustrated by Figs. 4.1 and 4.2, there exists a third row whose L' elements are all different from the $L'+1$ elements in the chain $(d_1, f_{i_1}, f_{i_2}, \ldots, f_{i_{L'-2}}, f_1, d_2)$, where $\{i_1, i_2, \ldots, i_{L'-2}\} = \{2, 3, \ldots, L'-1\}$.*

Proof. Any row beyond the first two rows of the G_8 table relates to A_1 by an offset $\Delta' \neq \Delta$. If d_1 or d_2 repeat in such a row then the G_8 table is not valid. With $\{i_1, i_2, \ldots, i_{L'-1}\} = \{1, 2, \ldots, L'-1\}$ and with $i_{L'-1} = 1$, we need to show that for a valid G_8 table that falls in Case 1, for any $k \neq \ell \in \{1, 2, \ldots, L'-1\}$, $f_{i_k} + \Delta' \neq f_{i_\ell}$ and $d_1 + \Delta' \neq f_{i_\ell}$.

Assume for a contradiction that $\exists\ k \neq \ell : f_{i_k} + \Delta' = f_{i_\ell}$. Define $n\Delta = \underbrace{\Delta + \Delta + \cdots + \Delta}_{n \text{ times}}$, where n is any non-negative integer. If n is negative, define $n\Delta = \underbrace{-\Delta - \Delta - \cdots - \Delta}_{-n \text{ times}}$. If $\ell > k$, then $f_{i_\ell} = f_{i_k} + (\ell - k)\Delta$ and hence $\Delta' = (\ell - k)\Delta$. Since $1 \leq \ell - k < L' - 1$ we can also obtain that $d_2 = f_{i_{L'-(\ell-k)}} + (\ell - k)\Delta$. This shows that d_2 would be an element of the new row, yielding a contradiction. The same argument in the opposite direction will show that d_1 will repeat as $d_1 = f_{i_{k-\ell}} + \Delta'$ if $\ell < k$. Similarly, one can show that, if $d_1 + \Delta' = f_{i_\ell}$ then d_2 will repeat. □

To summarize, there exist at least $(L' + 1) + L' + (L' - 3) = 3L' - 2$ distinct non-zero elements in a G_8 table that is valid and falls in Case 1, which means $N \geq 3L' - 1$: $L' + 1$ elements from $A_1 \cup \{d_2\}$, L' elements in a third row and the term $L' - 3$ appears from counting at least one distinct non-zero entry (on the diagonal) from each of the remaining rows.

4.4.2 Case 2

The situation where a single chain is not present within the set $A_1 \cup \{d_2\}$ is considered now. This is because, introducing only one new element when creating the second row from the first row, i.e. d_2, can also arise from the situation shown by the example in Fig. 4.3 (refer to the following description).

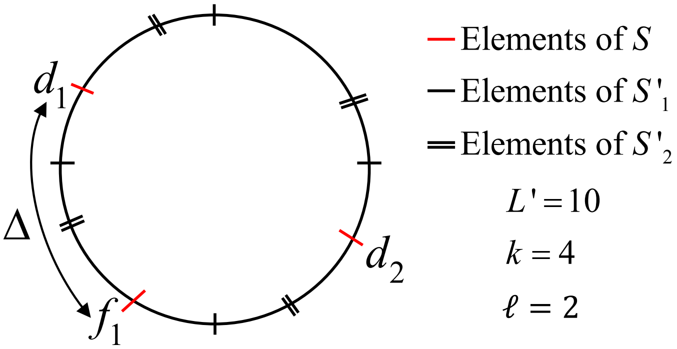

Figure 4.3: Alternative arrangement of elements of row 1 for Theorem 10

The elements of $A_1 \cup \{d_2\}$ could be in $\ell + 1$ disjoint sets:

1. Set S with elements $\{d_1, \underbrace{f_{i_1}, \ldots, f_{i_{L'-2-k\ell}}, f_1}_{L'-1-k\ell \text{ elements}}, d_2\}$.
2. ℓ other sets denoted $S'_j, 1 \leq j \leq \ell$ each comprising k elements such that within each set adding $n\Delta, n \in \mathbb{Z}$ to any element yields another element within the set itself.

While this case is introduced here, the proof that the theorem holds for it is given in the Appendix. This concludes the proof of Theorem 10. □

Although Theorem 10 only applies under specific constraints on the girth-8 table, we conjecture that the bound $N \geq 3L' - 1$ applies without these imposed constraints that $\exists\ i \neq j \in \{1, 2, \ldots, L'\}$ such that $|A_i \cap A_j|$ equals 0 or $L' - 1$.

4.5 Concluding Remarks

This chapter considered the problem of constructing $(3, L)$ quasi-cyclic low-density parity-check (LDPC) codes from complete protographs. An application of complete mappings from finite group theory provides explicit constructions of $(3, L)$ QC-LDPC codes that achieve girth $g = 6$ with the minimum possible lifting factor of L when L is odd. Identifying the minimum lifting factor required to obtain a girth of $g \geq 8$ is posed as a problem in additive combinatorics using the construct of a girth-8 table (G_8 table). An improved bound on the lifting factor is obtained under certain constraints on the cardinality of girth-8-table row-set intersections. We conjecture that this improved bound applies in general.

4.6 Appendix

4.6.1 Proof of Case 2

We prove here that Theorem 10 holds for Case 2 which was introduced in Section 4.4.2. The ℓ sets being referred to in Case 2 (from Fig. 4.3) have the same number of elements, denoted k here, or else adding Δ will create a new element for the second row, apart from d_2 which is already being created from $f_1 \in S$. To show that the theorem holds for this case, we focus on the $k\ell$ elements from the ℓ sets. For this case, a "linear" relationship within the elements of the $\ell + 1$ sets holds as follows. For the elements in S,

$$\begin{aligned} d_1 + \Delta =& f_{i_1}, \\ f_{i_1} + \Delta =& f_{i_2}, \\ &\vdots \\ f_{i_{L'-2-k\ell}} + \Delta =& f_1, \\ f_1 + \Delta =& d_2. \end{aligned}$$

For the elements in the ℓ sets S'_j,

$$\forall x \in S'_j, 1 \leq j \leq \ell, x + n\Delta \in S'_j, \forall n \in \mathbb{Z}. \tag{4.10}$$

Observe that if the elements from the group $\mathbb{Z}/N$ are chosen for A_1 according to Case 2, then the following holds:

$$\forall x \in \mathbb{Z}/N, x + k\Delta = x. \tag{4.11}$$

Each column of the G_8 table has a diagonal element that appears only once in the table. By adding offsets to the $k\ell$ elements in $(A_1 \cup \{d_2\}) \setminus S$ to obtain the $k\ell$ corresponding diagonal elements (and their respective rows), we have the following crucial observation.

Lemma 10. *For each column corresponding to an element in* $(A_1 \cup \{d_2\}) \setminus S$*, when obtaining a new diagonal element, at least* k *new non-zero elements are obtained in the corresponding row. Considering all* $k\ell$ *such rows, a total of* $k^2\ell$ *distinct elements, that are different from the elements in* $A_1 \cup \{d_2\}$*, is guaranteed for any valid* G_8 *table that falls under Case 2.*

Proof. Consider $x_1 \in (A_1 \cup \{d_2\}) \setminus S$ and assume that $x_1 \in S'_{i_1}$ for some $1 \leq i_1 \leq \ell$. There is an offset $\Delta_1 \neq \Delta$ such that $x_1 + \Delta_1 = d_{x_1}$, where d_{x_1} is the diagonal element in the column containing x_1. Note that the row containing d_{x_1} also contains every element in $S'_{i_1} + \Delta_1 = \{s + \Delta_1 : s \in S'_{i_1}\}$. No element in $S'_{i_1} + \Delta_1$ appears in $A_1 \cup \{d_2\}$ as this would force either $S'_{i_1} + \Delta_1 = S'_j, j \neq i_1$ so that $d_{x_1} \in A_1 \cup \{d_2\}$ or $S \subseteq S'_{i_1} + \Delta_1$ so that d_1 and d_2 appear in the row containing d_{x_1} due to (4.10). Either of these results would lead to a G_8 table that is not valid.

Now consider a second element $x_2 \in (A_1 \cup \{d_2\}) \setminus S$, $x_2 \neq x_1$ and $x_2 \in S'_{i_2}$, where $1 \leq i_2 \leq \ell$ is not necessarily different from i_1. There is an offset $\Delta_2 \notin \{\Delta, \Delta_1\}$ such that $x_2 + \Delta_2 = d_{x_2}$, where d_{x_2} is the diagonal element in the column containing x_2. Note that the row containing d_{x_2} also contains every element in $S'_{i_2} + \Delta_2$. Following the same reasoning as with x_1, no element in $S'_{i_2} + \Delta_2$ appears in $A_1 \cup \{d_2\}$. Also, $\left(S'_{i_1} + \Delta_1\right) \cap \left(S'_{i_2} + \Delta_2\right) = \emptyset$ or else $S'_{i_1} + \Delta_1 = S'_{i_2} + \Delta_2$ due to (4.10) and in particular $d_{x_2} \in S'_{i_1} + \Delta_1$ which would lead to a G_8 table that is not valid.

Continuing by induction yields $k^2\ell$ distinct elements that are not in the first row and are different from d_2. □

Thus we have for any valid G_8 table in Case 2 that

$$N \geq L' + 2 + k^2\ell, \tag{4.12}$$

where $L' + 2$ arises from counting the elements in $A_1 \cup \{d_2, 0\}$.

Lemma 11. *In the context of Case 2, where $|S| = L' - k\ell + 1$,*

$$L' - k\ell + 1 \leq k. \tag{4.13}$$

Proof. Due to (4.11). □

Case 2 is only possible when $L' \geq 5$, $k \geq 3$ and $\ell \geq 1$. We consider two ranges for k as follows: If $k \geq \sqrt{2L' - 3}$, then

$$\begin{aligned} N \geq & L' + 2 + k^2 \ell \\ \geq & L' + 2 + k^2 \\ \geq & L' + 2 + 2L' - 3 = 3L' - 1. \end{aligned} \tag{4.14}$$

If $k < \sqrt{2L' - 3}$, we first use (4.13) in (4.12) to get

$$\begin{aligned} N \geq & L' + 2 + k^2 \ell \\ \geq & L' + 2 + kL' - k^2 + k \end{aligned} \tag{4.15}$$

which yields a quadratic expression in k for every L'. This is concave in k and it can be verified that the maximum of the right-hand side is obtained at $k_{\max} = \frac{L'+1}{2}$. Under the supposition that $k < \sqrt{2L' - 3}$, we can also trivially verify that $k < \sqrt{2L' - 3} < k_{\max}$ for $L' \geq 5$ and thus to minimize the right-hand side, we have to set k to the smallest feasible value, which is $k = 3$. This yields

$$\begin{aligned} N \geq & L' + 2 + 3L' - 9 + 3 \\ = & 4L' - 4 > 3L' - 1, \forall L' \geq 5, \end{aligned} \tag{4.16}$$

which completes the proof for Case 2 and thus of Theorem 10.

Acknowledgment

The material in this chapter is based upon work that was supported by the Broadcom Foundation and NSF grants CCF-1162501 and CCF-1161822. Research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA and JPL-NSF Task Plan 82-17473.

CHAPTER 5

Allocating Redundancy Between Erasure Coding and Channel Coding when Fading Channel Diversity Grows with Codeword Length

This chapter[1] turns towards communication theory and tackles a rate allocation problem studied previously in literature, but with an important twist. Consider a cross-layer coding scheme with packet-level erasure coding and physical-layer channel coding. It is known from previous work that some erasure coding is necessary even in the limit of large physical-layer codeword block-lengths if the physical-layer fading channel does not provide diversity that grows with block-length. However, is erasure coding still required in the limit of large block-lengths if the physical layer allows for diversity to grow with block-length? The theoretical answer turns out to be a resounding “no” in the case of Rayleigh fading that allows diversity to increase linearly with block-length.

5.1 Introduction

A transmitter desires to communicate a message composed of a certain number of information packets over a fading channel. It uses an inter-packet erasure code to produce erasure-

[1] Material in Section 5.6 of this chapter was presented at the 2016 International Zurich Seminar on Communications (IZS) [WVR^{+}16]. The chapter appeared (subject to minor differences) as a journal article in IEEE Transactions on Communications [RMW17]. Tong Mu, of Stanford University, collaborated on work that resulted in this chapter when she was at UCLA.

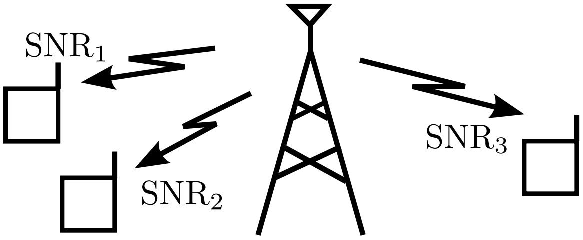

Figure 5.1: Wireless broadcasting to heterogeneous receivers

coded packets and then transmits each resultant packet using a physical-layer channel code. Throughout the chapter, the term codeword refers to the channel codewords that are transmitted, one for each erasure-coded packet. The receiver decodes each received codeword and then recovers the overall message by decoding the erasure code. Coding schemes such as this are generally categorized under cross-layer designs.

Consider the wireless communication system of Fig. 5.1 in which incremental redundancy from the transmitter gradually lowers the rate of the transmission until the receiver(s) can successfully decode the packetized message. A coding scheme such as the one described above is practically useful in such wireless applications [Tec15] and might be implemented using, for instance, "rateless" erasure codes such as Raptor codes [Sho06] with powerful channel codes such as low-density parity-check (LDPC) codes [Gal63]. Additionally, in the context of a wireless broadcast, an optimization goal is to design the stream of incremental redundancy so that each receiver in the broadcast may decode all the packets in the message as early as possible, given its signal-to-noise ratio. To accomplish this, for each overall code rate an optimization objective can be to minimize the value of the average signal-to-noise ratio at which successful decoding of the whole message is possible at the receivers.

Optimization problems for cross-layer schemes and the results of the optimization depend upon the channel model and network configuration under consideration. The importance of cross-layer designs in the context of communication over a fading channel is well established by Luby et al. [LGSW07] who show that systems without a packet-erasure code are highly suboptimal when packets are sent over fading channels.

Courtade and Wesel [CW11], in their work on cross-layer design, summarized a large portion of literature in optimizing cross-layer schemes including four notable works: An early work is that of Vehkapera and Medard [VM05], where the authors maximize throughput under an overall code-length criterion for a non-fading channel. Xiao et al. [XMA11] consider cross-layer optimization to minimize the expected delay of a network-coded system over a non-fading physical layer. Berger et al. [BZW$^+$08] consider cross-layer optimization of rates over a fading channel with the fade value remaining constant over each channel codeword. At the other end of the spectrum, Cui et al. [CHK09] consider a fading channel wherein each channel codeword symbol faces an independent fade. Apart from these, Cao and Blostein [CB10] consider optimizing over a discrete set of overall rates. The authors examine both slow and fast fading in this simplified and pragmatic version of the problem.

More recent literature has focused on studying various other physical-layer channel models and multi-user scenarios. These works include Sun's work [Sun11] on a relay-aided system employing network coding and channel coding. Here, the author solves the redundancy-allocation problem using the block-fading model, wherein each codeword faces only one block fade irrespective of its block-length, similar to Berger et al. [BZW$^+$08]. Guo et al. [GHW$^+$12] study a scheme in a practical setting using LDPC codes, demonstrating the benefit of such a scheme over others. Koller et al. [KHKC11], [KHKC14] also study network-coded unicast and broadcast systems, but over the binary symmetric channel. They find that additional redundancy can be more beneficial when used to produce longer channel codewords as opposed to sending additional network-coded packets.

Beyond cross-layer designs such as the one considered in this chapter and in the works referred to so far, recent works have also focused on efficient cross-layer code designs that do not treat codewords that are not decoded successfully as erasures. Instead, these schemes supply additional redundancy to codewords that are not successfully decoded and facilitate further decoding attempts. These works include the inter-frame coding approach of Zeineddine and Mansour [ZM16] and the packet-combining scheme proposed by Lin et al. in [LXXL11].

5.1.1 Contributions

This chapter considers transmission of a packetized message using the cross-layer coding scheme that combines inter-packet erasure coding and physical-layer channel coding (as explained in Section 5.1). Let the probability that the receiver fails to decode the message be called the message error probability. Given an overall rate, we formulate an optimization problem to minimize the signal-to-noise ratio (SNR) required to achieve a certain message error probability by optimizing the redundancies allocated to the two component codes. This chapter extends Courtade and Wesel's work [CW11] on minimizing the power for transmitting a packetized message over a fading channel.

Proportional diversity

The initial work [CW11] assumes a block-fading channel with a constant number of fades per codeword, irrespective of the block-length of the codeword. In this chapter, we assume that the number of fades per codeword grows linearly with the codeword length. This assumption generalizes the work of [CW11], allowing us to analyze systems with time-varying fading (e.g. from mobility) where the coherence time of fading naturally remains a constant irrespective of the codeword length.

Outage approximations for non-integer number of fades

The proportional-diversity model introduces the complication of a non-integer number of fades when the codeword ends before the final segment of fading completes. To handle this, we develop two Gaussian approximations to the outage probability and use Monte-Carlo evaluations to demonstrate that the approximations are a good fit and thus can be used to obtain numerical results efficiently.

Analytical proof that erasure code rate converges to one

We provide an analytical proof that for any system where diversity grows with codeword length the optimal erasure code rate is equal to its maximum possible value (e.g. a rate of 1 for an erasure code with no overhead) for the lowest overall code rates. This is in contrast to the result in [CW11] for systems with a fixed value of diversity (i.e. no mobility) where the optimal erasure code rate decreases to a limiting value less than its maximum allowable value. Thus, [CW11] shows that erasure coding is beneficial even at lowest overall code rates for a system with fixed diversity, whereas our result shows that with time-varying fading that allows diversity to increase with codeword length, erasure coding provides no benefit at the lowest overall rates. The approach in our analytical proof of the optimal value of the optimization problem at low overall rates is completely different from any analysis in [CW11].

Numerical characterization

By numerically solving the new optimization problem over a wide range of overall rates, we show that the optimal erasure code rate might decrease at first with decreasing overall rate but then remains roughly constant for a large rate range before climbing up again for very low values of overall code rate. For systems where diversity within a transmitted codeword grows quickly with the codeword length, the optimal erasure code rate remains constant for most overall code rates of interest before increasing to its maximum possible value for the lowest overall rates.

A related optimization problem

In addition to considering the optimization framework of [CW11] in which the transmission power is minimized under the constraint of a fixed message error probability, this chapter also considers an alternative framework in which the probability of message error is minimized with a constant transmission power.

Remark 9. *An important contribution of this chapter is the insight it provides for systems with rate-compatibility. The mechanics of each instance of the optimization assume a fixed value of the overall rate. However, the central insights of our work come from examining how these optimization results evolve as the overall rate decreases, as would happen in a system with incremental redundancy. Such systems are often implemented with rate-compatible codes that allow redundancy to be added to either the physical-layer channel code or the packet-level erasure code or both. This chapter makes observations that guide the design choice regarding where to place the incremental redundancy in systems with time-varying fading, e.g. systems with mobility.*

5.1.2 Organization

The chapter is organized as follows: Section 5.2 introduces notation. Section 5.3 sets up the optimization problem and provides the approximations required to numerically solve the problem. Section 5.4 proves that the optimal erasure code rate is equal to the maximum possible erasure code rate value for any set of system parameters if the overall code rate is small enough. Section 5.5 presents the results of numerically solving the optimization problem and discusses the results. Section 5.6 discusses the related optimization problem of minimizing the message error probability at a fixed transmit power. Section 5.7 concludes the chapter.

5.2 Preliminaries and Notation

The notation followed here is the same as in [CW11]. Consider the complex fading channel

$$Y = HX + N \tag{5.1}$$

in which X is the transmitted symbol, H is the fading coefficient, and Y is the received symbol. The average transmit power is P, and the variance of the zero-mean, complex and circularly-symmetric Gaussian noise, N, is 1. The receiver has perfect knowledge of the fading random variable H. The fading parameter H is Rayleigh and $\mathbb{E}[HH^*] = 1$. For this Rayleigh fading channel, SNR at the receiver, denoted γ, is exponentially distributed with parameter $\frac{1}{P}$. The channel capacity [nats/channel use] for an instantaneous SNR γ is

$$C(\gamma) = \ln(1+\gamma). \tag{5.2}$$

A message composed of m packets with k nats of information per packet is to be transmitted over this fading channel. The probability that the receiver fails to decode the message (i.e. fails to decode one or more of the packets that comprise the message) is denoted by q. Let there be T channel uses for an overall rate of mk/T [nats/channel use]. The transmitter employs an erasure code of rate R_E across the m packets and transmits each of the $\frac{m}{R_E}$ erasure-coded packets using a channel code of rate R_C [nats/channel use] so that

$$\frac{mk}{T} = R_E R_C. \tag{5.3}$$

That is, there are $\frac{m}{R_E}$ erasure-coded packets that are obtained from the m message packets using the erasure code, and each channel codeword, of which there are $\frac{m}{R_E}$ in total, is $\frac{k}{R_C}$ symbols long. Note that R_E has to satisfy $R_E \leq 1$. We constrain $\frac{m}{R_E}$ to be an integer throughout the chapter. We do not have such a constraint on $\frac{k}{R_C}$, but such a constraint would not change anything that follows except for reducing further the feasible set for the optimization problem.

The receiver decodes the erasure code and recovers the message successfully whenever the channel decoder correctly decodes a sufficient number of the $\frac{m}{R_E}$ codewords. The number of packets that the decoder of the erasure code requires to recover the message, denoted $\hat{m} \geq m$, depends upon the erasure code. For Reed-Solomon erasure codes, $\hat{m} = m$; for fountain codes such as a Raptor code, $\hat{m} > m$ typically. Thus, $\hat{m}$ is the fixed number of packets that the decoder of the erasure code requires and $R_C T k^{-1} = \frac{m}{R_E} \geq \hat{m} \geq m$.

5.2.1 Proportional-Diversity (PD) Block-Fading Model

A common block-fading model, found for example in Goldsmith [Gol05] and Biglieri [Big05], assumes that the number of fades per codeword is a constant irrespective of the codeword length. This model applies, for example, to a static OFDM channel where the number of fades is controlled by the coherence bandwidth of the frequency selective channel. Many previous works on cross-layer optimization, which are referred to in Section 5.1, assume this model. Another possible model, which we refer to as *proportional-diversity block fading (PD block fading)* fixes the number of channel symbols (i.e. the fade length) in a codeword that encounter the same fade value $H = h$. This model applies, for example, to a time-varying fading channel where the fade length is controlled by the coherence time. We use the parameter l_f to denote the fade lengths. With the block-length being $\frac{k}{R_C}$, the number of block fades F in a transmitted codeword of a system with PD block fading with fade lengths l_f is

$$F = \frac{k}{R_C l_f}. \tag{5.4}$$

The block fade values H are i.i.d. with respect to each other. The value F represents the total fractional number of fades in the codeword, which is typically not an integer. The integer number of distinct fades in the codeword is $\lceil F \rceil$. One way of understanding the difference between [CW11] and the current chapter is that in [CW11] F is a constant fixed integer value, whereas in the current chapter F is a real number that varies as a function of R_C according to (5.4).

In this chapter, we assume that k and l_f are fixed constants known to both the transmitter and the receiver. The assumption that l_f is fixed is natural for time-varying channels where the coherence time of fading remains constant and the symbol transmission rate is chosen to be constant. For the probability of successfully decoding a codeword to exceed a specified target, the transmitter must use a codeword length that can harvest a sufficient amount of diversity. For that subset of cases where F is an integer, the optimization problem solved in [CW11] can be adapted to obtain the optimal erasure and channel code rates required in the new PD framework. However, given a fixed k and l_f, constraining R_C so that F is always an integer significantly restricts the set of values of R_C that can be employed in the system. For this reason, the fading model employed in this chapter both generalizes the work in [CW11] and also does not restrict the values of R_C that can be chosen for a system that would employ the two-layer coding scheme.

5.3 The Optimization Problem

Given an overall rate of $R_E R_C = mk/T$, the transmitter transmits $\frac{m}{R_E} = R_C T k^{-1}$ erasure-coded packets, each at a channel code rate of R_C. We assume that the channel code for every erasure-coded packet is the same. The message error probability q, which is the probability that the erasure decoder at the receiver fails to acquire $\hat{m} \geq m$ packets, is approximated as in [CW11] using the normal approximation to a binomial probability:

$$q \approx \Phi\left[\frac{(\hat{m}-1) - R_C T k^{-1}(1-p_e)}{\sqrt{R_C T k^{-1} p_e (1-p_e)}}\right]. \tag{5.5}$$

Here, p_e is the probability that the receiver cannot decode a channel codeword (which is seen as an erasure event by the erasure code), and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. To be more precise, the binomial probability that is approximated in (5.5) is the probability that the receiver fails to acquire $\hat{m}$ unerased packets out of a total of $R_C T k^{-1}$ erasure-coded packets given that the erasure events (i.e. unsuccessful channel-code decoding) are independent for each packet. The approximation is tight when $\hat{m}$ is sufficiently large and we assume reasonably large values ($\hat{m} = 32$, 64) for our purposes. Let λ be the allowable message error probability for the transmission. That is, $q \leq \lambda$ must be satisfied.

With the notation and terminology developed above, we define the optimization problem that we solve in this chapter as follows: For specified k, T, m, $\hat{m}$, and l_f,

$$\begin{aligned} &\min_{R_C} \quad P \\ &\text{s.t.} \quad q \leq \lambda, \\ R_C > 0 \text{ satisfying } &\hat{m} \leq \frac{m}{R_E}, \ \frac{m}{R_E} \in \mathbb{Z}. \end{aligned} \tag{5.6}$$

Note that the overall rate mk/T is fixed by the specified parameters and that R_E is obtained as $R_E = mk/(TR_C)$. Thus the optimization allocates the overall rate between R_C and R_E to minimize P while maintaining a message error probability less than λ. The message error probability q, as described in (5.5), is a function that depends on R_C both explicitly and through the variable p_e. We begin by describing our model for p_e and its dependence on the parameters P, R_C, k, and l_f.

5.3.1 Probability of Codeword Decoding Error

The modeling and computation of p_e are the most intricate parts of the optimization problem. We model this probability that the channel decoder fails to decode a channel codeword using the outage probability of the fading channel and by accounting for finite-length effects of channel coding. For our PD block-fading channel model with a fade length of l_f, we model p_e as the probability of the following event:

$$\left\{ \frac{\sum_{i=1}^{\lfloor F \rfloor} C\left(\gamma_i\right) + \left(F - \lfloor F \rfloor\right) C\left(\gamma_{\text{last}}\right)}{F} < (1+\epsilon) R_C \right\}, \tag{5.7}$$

where F is defined in (5.4). This event is the outage event for PD block fading when the channel code rate is R_C. The random variables in this expression are the values $C(\gamma_i)$, one for each fade. We account for the limitations of finite-length channel codes through the small constant $\epsilon > 0$ as practical channel codes must operate at rates below capacity. Also, it is assumed for simplicity that every codeword begins with a new fading block of length l_f even if the codeword preceding it ended before a block fade of length l_f completed.

The outage probability for the regular block-fading channel can be computed using its exact closed-form expressions as shown, for example, by Yilmaz in [Yil11]. However, we do not have an exact closed-form expression to evaluate p_e (the probability of (5.7)) for our PD block-fading model. We therefore approximate p_e. It turns out that two approximations that we provide below are almost equal to each other, and are both quite close to the true p_e when the probability of the event in (5.7) is evaluated using a Monte-Carlo simulation.

Let us first rewrite (5.7) using the substitutions $c = (1 + \epsilon)$, $W_i = \ln(1 + \gamma_i)$, $W_{\text{last}} = \ln(1 + \gamma_{\text{last}})$ and simplifying:

$$p_e = \mathbb{P}\left[\sum_{i=1}^{\lfloor F \rfloor} W_i + (F - \lfloor F \rfloor) W_{\text{last}} < \frac{ck}{l_f}\right]. \tag{5.8}$$

The mean and variance of W_i (which are all i.i.d.) depend on the transmitter power P, so we will refer to these as $\mu(P)$ and $\text{Var}(P)$. These nonnegative constants, which will be used in this chapter throughout, can be computed as shown in the work by Courtade and Wesel [CW11]:

$$\mu(P) = e^{1/P}\alpha(P), \tag{5.9}$$

$$\text{Var}(P) = 2e^{1/P}\beta(P) + 2e^{1/P}\ln(P)\alpha(P) - e^{2/P}\alpha^2(P), \tag{5.10}$$

where

$$\alpha(P) = \int_{P^{-1}}^{\infty} \frac{1}{t} e^{-t} dt, \quad \text{and} \tag{5.11}$$

$$\beta(P) = \int_{P^{-1}}^{\infty} \frac{\ln(t)}{t} e^{-t} dt. \tag{5.12}$$

Note that the integrals required to obtain $\alpha(P)$ and $\beta(P)$ are difficult to compute for a very low value of P [CW11], but our numerical results exploring the effect of the diversity growth factor in Section 5.5 are not in the low-SNR regime and hence this is not an issue.

Now, let us assume that $\sum_{i=1}^{\lfloor F \rfloor} W_i$ is a Gaussian random variable and also assume that

$(F - \lfloor F \rfloor) W_{\text{last}}$ is a Gaussian random variable. The sum of these two Gaussian terms is thus another Gaussian, W_G, which stands for *Gaussian approximation of weighted average mutual information.* Let $\text{mean}(W_G)$ and $\text{Var}(W_G)$ be the mean and variance of the overall Gaussian random variable. We can now obtain the following Gaussian approximation to p_e by evaluating the cumulative distribution function of a standard normal distribution:

$$\begin{aligned} p_e &\approx \Phi\left[\frac{\frac{ck}{l_f} - \text{mean}(W_G)}{\sqrt{\text{Var}(W_G)}}\right], \text{ where} \\ \text{mean}(W_G) &= F\mu(P), \\ \text{Var}(W_G) &= \text{Var}(P)\left[\lfloor F \rfloor + (F - \lfloor F \rfloor)^2\right]. \end{aligned} \tag{5.13}$$

This approximation can be evaluated by computing $\mu(P)$ and $\text{Var}(P)$ and evaluating $\Phi(\cdot)$.

A simpler approximation is to use the Gaussian approximation to the outage probability as if F were an integer in (5.7) even with non-integer values of F. This results in (5.14). When $F < 1$ (i.e. $R_C > \frac{k}{l_f}$), the event in (5.7) evaluates to $\{C(\gamma) < (1+\epsilon)R_C\}$ since there is only one fade in the whole codeword. Hence we obtain

$$p_e \approx \begin{cases} \Phi\left[\sqrt{F}\frac{cR_C - \mu(P)}{\sqrt{\text{Var}(P)}}\right], & F \geq 1 \\ \Phi\left[\frac{cR_C - \mu(P)}{\sqrt{\text{Var}(P)}}\right], & F < 1 \end{cases}. \tag{5.14}$$

Note that the approximation in (5.13) is equal to the approximation in (5.14) when $F < 1$, i.e. when $R_C > \frac{k}{l_f}$. Equation (5.14) is the key approximation that we will use in the rest of this chapter. Its accuracy will be justified via Figs. 5.2 and 5.3. Moreover, we prove in Appendix 5.8.1 that (5.14) is a well-behaved decoding error probability function of R_C. That is, the decoding error rate function for a codeword increases with the channel code rate R_C. This requirement is informally clear from (5.8): the left-hand side of the event in (5.8) decreases in the number of random variables in it as R_C increases and therefore the probability that the weighted average mutual information is less than $\frac{ck}{l_f}$, which is a constant,

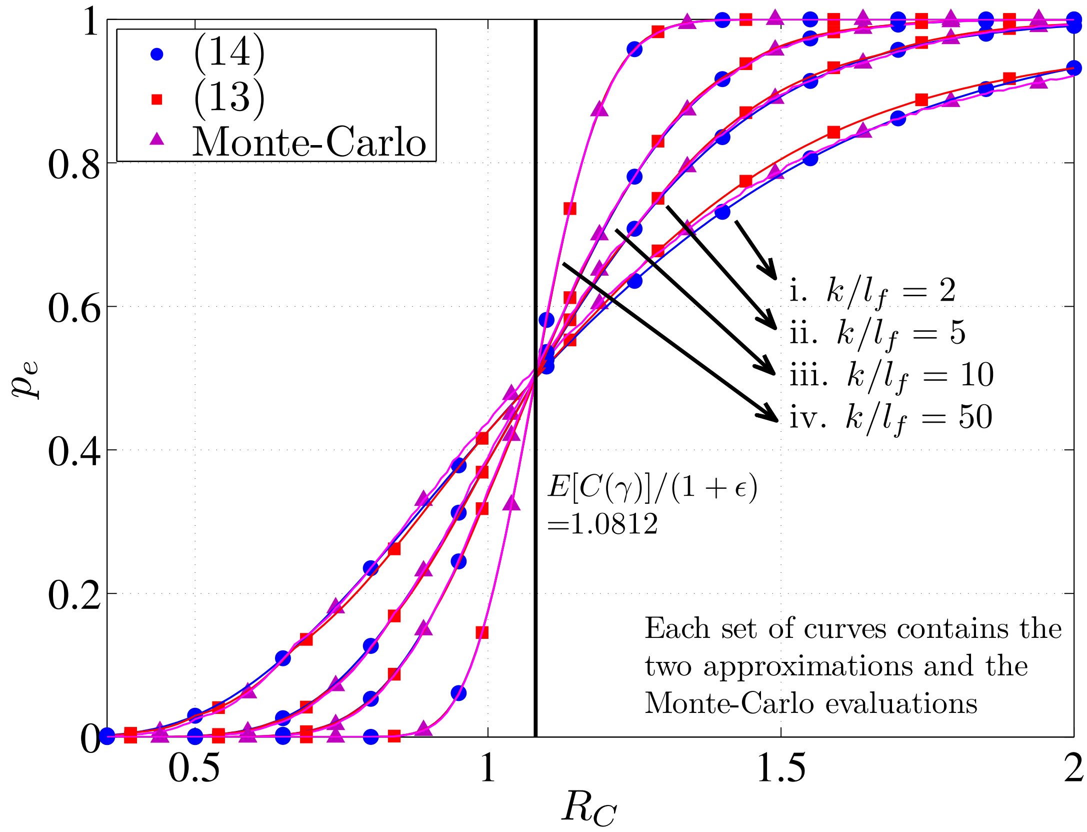

Figure 5.2: Comparison of the two Gaussian approximations of p_e with its Monte-Carlo evaluation at 5dB. Also indicated in the figure is the value of $\mathbb{E}[C(\gamma)]/(1+\epsilon)$ for 5 dB, which was computed using a numerical integration to find the expectation.

increases as R_C increases.

Figs. 5.2 and 5.3 show the evaluation of the two approximations (5.13) and (5.14) and the result from the Monte-Carlo evaluation of (5.8) for $\epsilon = 0.1$, $P = 5$ and 10 dB, and various values of $\frac{k}{l_f}$. The approximations closely follow the Monte-Carlo results.

Consider the behavior of p_e with respect to the ratio $\frac{k}{l_f}$, which we call the *diversity growth factor (DGF)*. As DGF increases, the probability of outage p_e approaches a step function. This is because, looking at the equation in (5.7), the left-hand side is essentially an average of roughly $\frac{k}{R_C l_f}$ random variables that are i.i.d. and of the form $C(\gamma)$. As $\frac{k}{l_f} \to \infty$, by the law of large numbers, the left-hand side converges to the expectation of $C(\gamma)$, which is actually the ergodic capacity of Rayleigh fading channel. In this limit, the outage event in (5.7) thus

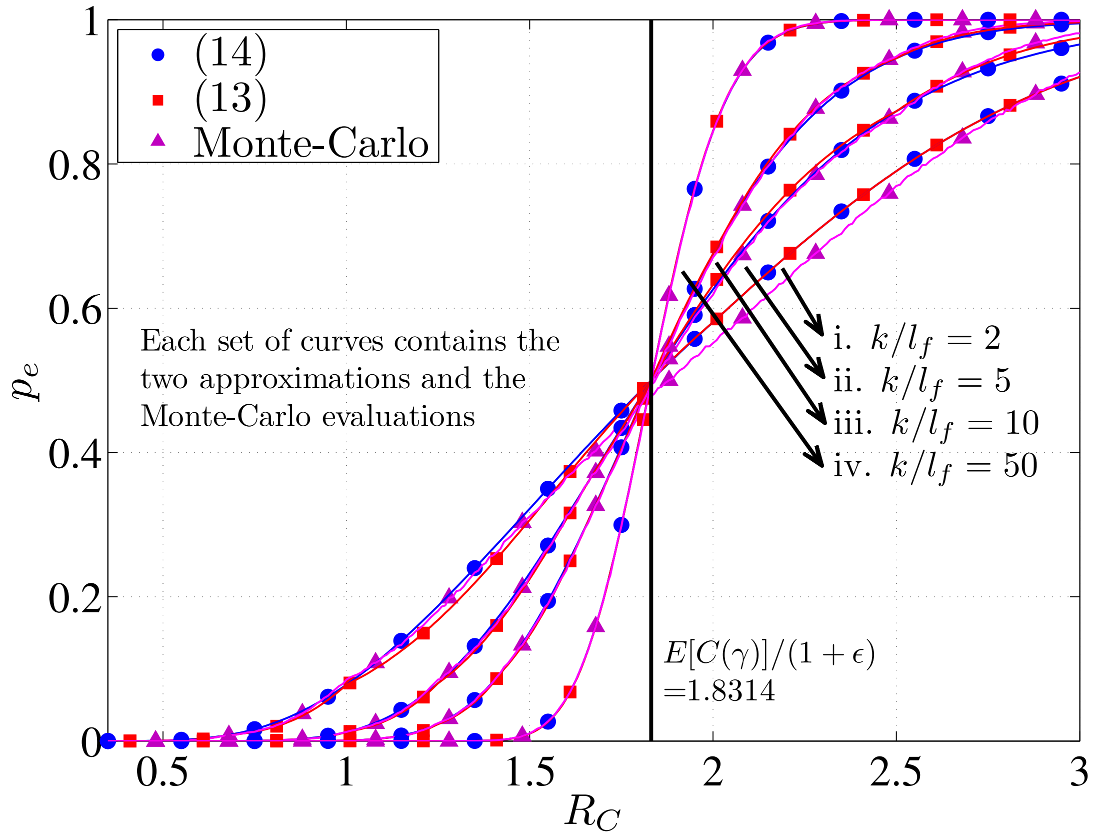

Figure 5.3: Comparison of the two Gaussian approximations of p_e with its Monte-Carlo evaluation at 10dB

becomes

$$\left\{\frac{\mathbb{E}[C(\gamma)]}{1+\epsilon} < R_C\right\}. \tag{5.15}$$

This means that in the limit as $\frac{k}{l_f} \to \infty$, p_e converges to a step function with the step occurring at $\frac{\mathbb{E}[C(\gamma)]}{1+\epsilon}$. This value is indicated as a black vertical line in Figs. 5.2 and 5.3 for the two power values 5 dB and 10 dB. Since the two approximations and the Monte-Carlo evaluation of (5.8) are almost identical, we use (5.14) to evaluate (5.8) in the sequel.

5.3.2 Numerical Solution to the Optimization Problem

The optimization problem given in (5.6) can thus be re-written via the discussion so far as

$$\begin{aligned} &\min\nolimits_{R_C} \quad P \\ &\text{s.t.} \;\; \Phi\left[\frac{(\hat{m}-1) - R_C T k^{-1}(1-p_e)}{\sqrt{R_C T k^{-1} p_e (1-p_e)}}\right] \le \lambda, \\ &\frac{\hat{m}k}{T} \le R_C, \frac{m}{R_E} \in \mathbb{Z}. \end{aligned} \tag{5.16}$$

The lower bound to R_C in the second constraint in (5.16) is due to $\frac{\hat{m}k}{T} \le \frac{mk}{TR_E} = R_C$. Following Courtade and Wesel in [CW11], the optimization problem is equivalent to (using $F = \frac{k}{R_C l_f}$)

$$\begin{aligned} &\min\nolimits_{R_C} \quad P \\ &\text{s.t.} \;\; p_e^*(R_C) = \begin{cases} \Phi\left[\sqrt{\frac{k}{R_C l_f}} \frac{cR_C - \mu(P)}{\sqrt{\mathrm{Var}(P)}}\right], & R_C \le \frac{k}{l_f} \\ \Phi\left[\frac{cR_C - \mu(P)}{\sqrt{\mathrm{Var}(P)}}\right], & R_C > \frac{k}{l_f} \end{cases}, \\ &\frac{\hat{m}k}{T} \le R_C, \frac{m}{R_E} \in \mathbb{Z}, \end{aligned} \tag{5.17}$$

where $p_e^*(R_C)$ is the value of channel codeword decoding error probability that leads to equality in the message error probability constraint $q \le \lambda$ in (5.16). This value is obtained as derived in equations (12)-(13) in [CW11].

Thus, numerically solving the optimization problem involves the following steps: First, for any fixed R_C we obtain $p_e^*(R_C)$ that satisfies $q \le \lambda$ in (5.16) with equality. For this R_C, the value of P that satisfies the first constraint in (5.17) with equality is obtained through a bisection search in P of the right-hand side of the constraint. Then, we consider all possible values of R_C such that $\frac{m}{R_E} \in \mathbb{Z}$ and find the value of R_C that provides the smallest value of P yielding a message error probability equal to the maximum allowable value λ.

We refer the reader to Section III of [CW11] for the derivation of the equations (12)-(13) of [CW11] that allow computation of the value of $p_e^*(R_C)$, but we provide those equations from [CW11] for convenience below:

Let $\lambda' = -\Phi^{-1}(\lambda) > 0$ and $\lambda'' = (\lambda')^2$. Equation (12) of [CW11] is the following equation in Z:

$$\begin{aligned} & Z^2 R_C T k^{-1} \left(R_C T k^{-1} + \lambda'' \right) \\ & - Z R_C T k^{-1} \left(2(\hat{m}-1) + \lambda'' \right) + \left(\hat{m}-1\right)^2 = 0 \end{aligned} \tag{5.18}$$

This equation is solved to find the larger root Z_1. After this, (13) of [CW11] obtains $p_e^*(R_C)$ as follows:

$$p_e^*(R_C) = 1 - Z_1. \tag{5.19}$$

Note that the change from fixed diversity in [CW11] to proportional diversity in the current work does not affect the derivation of $p_e^*(R_C)$ in (5.18) and (5.19) above. However, the introduction of proportional diversity to the block-fading channel does change the right-hand side of (5.17) from the corresponding expression in [CW11] by replacing the constant term F in (19) of [CW11] with the fractional number of fades $F = \frac{k}{R_C l_f}$ in (5.17). Note that $\frac{k}{R_C l_f}$ increases with decreasing R_C, which (as we see below) changes the behavior of the resulting optimal values of R_C and R_E.

For five example values of overall rate, Fig. 5.4 shows the objective function as a function of $1/R_E$ for a range of possible (R_E, R_C) pairs that support the specified overall rate. A star is used to identify the (R_E, R_C) pair that optimizes the objective function for a specified overall rate. Note that as T increases, the curve of the objective (minimum transmit power) vs. $1/R_E$ flattens so that a larger range of R_E values provide optimal or near-optimal performance.

We note that the model for probability of decoding error for a codeword (5.7) loses accuracy when the number of information bits in a message packet is small (on the order of a few hundred bits), as shown in the recent work on finite block-length information theory for Rayleigh fading channels by Durisi et al. [DKO$^+$16]. In this chapter, the optimization problem (5.16) depends upon only the ratios $\frac{k}{l_f}$ and $\frac{k}{T}$ and not explicitly on k. Therefore, a limitation of this chapter is that the results of the optimization problem may not be accurate

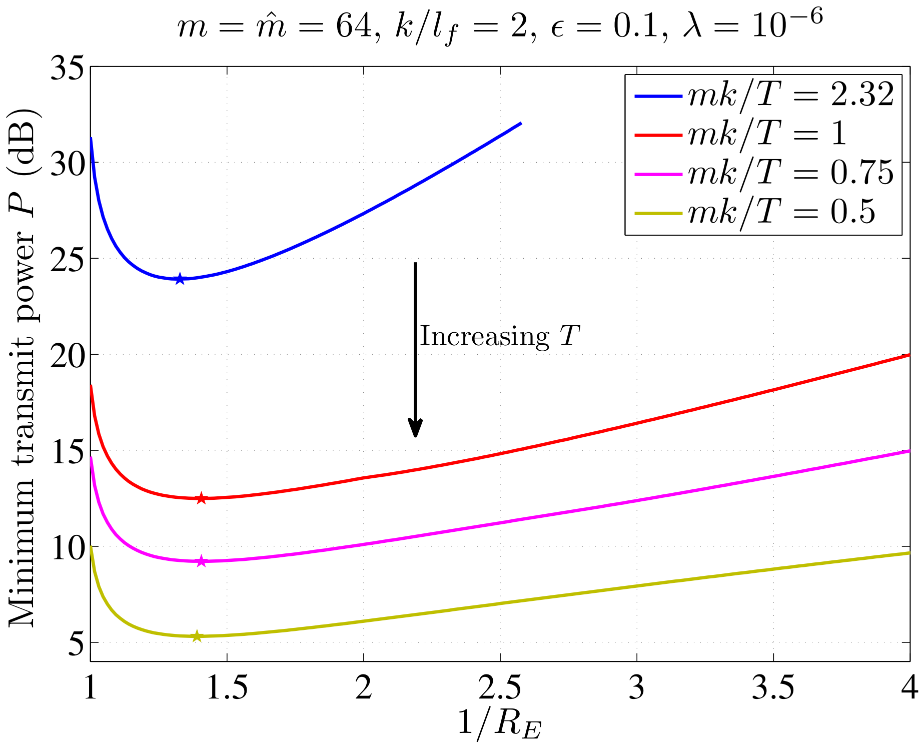

Figure 5.4: Value of the objective function of the optimization problem at different feasible R_E, R_C pairs in the feasible set

if used in the context of small values of k. For such small values of k, the accuracy issue can be addressed by tuning ϵ as a function of k according to guidance from works such as [DKO+16].

5.4 Analysis of the Optimal Value of the Erasure Code Rate at Low-Enough Overall Rates

In this section, we show analytically that the optimal erasure code rate is equal to $\frac{m}{\hat{m}}$ (effectively meaning that there is no erasure coding) for any sufficiently low overall code rate regardless of the value of all other system parameters.

Theorem 11. *Let the coding scheme use an arbitrary erasure code that requires $\hat{m} : m \leq \hat{m} \leq \frac{m}{R_E}$ packets to decode the overall message. Let us assume a Rayleigh proportional-diversity block-fading channel of fade lengths l_f. Further, let the feasible set of R_C values* *be constrained to* $R_C \in \left[\frac{\hat{m}k}{T}, \frac{k}{l_f}\right]$. *Then, the optimal value of* R_E, *which is a function of* T, *is equal to* $\frac{m}{\hat{m}}$ *for any sufficiently large* T *given any set of parameters* m, $\hat{m}$, k, l_f, λ, *and* $c = 1 + \epsilon$.

Proof. The proof is given in Appendix 5.8.2. We provide a proof sketch here as follows: For a given T, finding the power level P that achieves the target message error probability λ for a fixed R_C first requires finding $p_e^*(R_C)$ as described in Section 5.3.2. Then the transmitter power level P that yields this $p_e^*(R_C)$ as the probability of codeword decoding error is the smallest transmit power that satisfies $q \leq \lambda$. According to these steps, we find $p_e^*(R_C)$ for the lowest possible channel code rate $R_C = \hat{m}k/T$ and then the required transmitter power level $P_0(T)$ that yields $p_e^*(R_C = \hat{m}k/T)$. We then show that for any value of T that is large enough, the power level $P_0(T)$ cannot satisfy $q \leq \lambda$ for any other value of R_C, which implies that $R_C = \hat{m}k/T$ is the value that provides the smallest transmit power and hence the optimal R_E is $\frac{m}{\hat{m}}$. □

Remark 10. *We note that Theorem 11 is not only an asymptotic result as* $T \to \infty$, *but rather a result that is also valid for all low-enough overall code rates* mk/T.

Remark 11. *The statement of the theorem imposes a technical constraint of* $R_C \leq \frac{k}{l_f}$. *While our proof technique does not naturally handle the case when there is no such upper bound on* R_C, *our empirical results indicate that the optimal value of* R_C *decreases as* T *increases for large* T. *Thus the upper bound, while necessary for the proof, does not appear to be practically important.*

5.5 Numerical Results and Discussion

This section presents the results of numerically solving the optimization problem (5.17). For our results we set $\epsilon = 0.1$ and assume that $\hat{m} = m$, which is the case for Reed-Solomon codes. Our target message error probability λ is 10^{-6}. The optimizations were carried out using

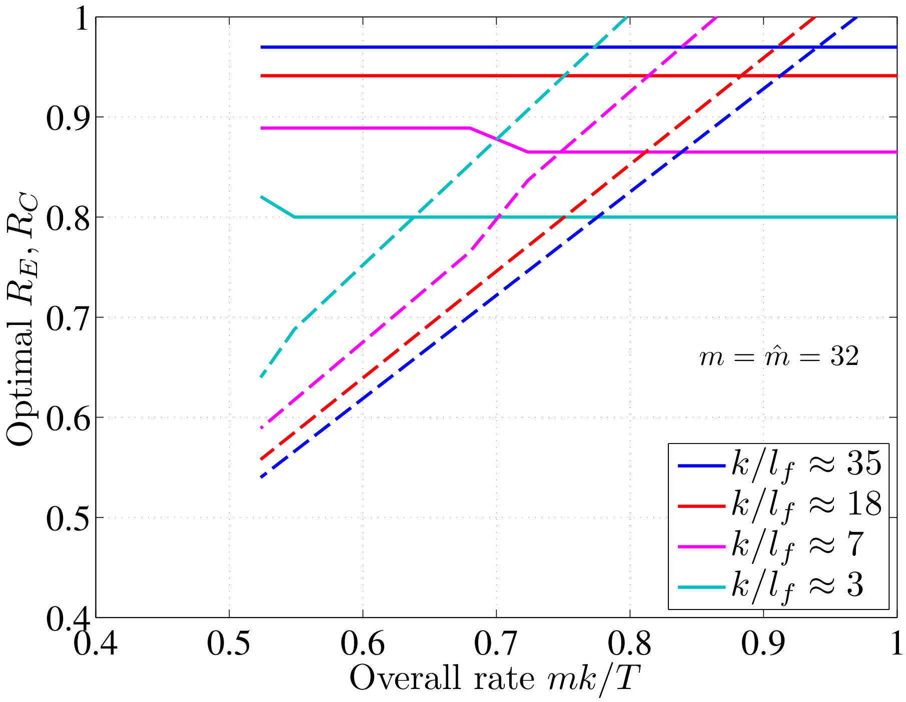

Figure 5.5: Optimal component code rates for systems with allowable message error probability λ of 10^{-6}, $\hat{m} = m$, and $\epsilon = 0.1$. Dashed lines represent optimal R_C and solid lines represent optimal R_E.

a MATLAB script. We present results separately for moderate overall rates ($mk/T > 0.5$) and for very low overall rates ($mk/T < 0.15$).

5.5.1 Moderate Overall Rates

Fig. 5.5 presents example results of the optimization routine for message consisting of $m = 32$ packets. We consider four values of $\frac{k}{l_f}$ here: $35, 18, 7, 3$. The optimal R_E value is roughly constant across the overall code rates considered here, and the optimal R_C is approaching zero monotonically with decreasing overall rates. The results in Fig. 5.5 can be interpreted in the following way if we start with a particular overall rate for the system and then employ incremental redundancy: It is beneficial to use the additional incremental redundancy to the physical-layer channel coding of each erasure-coded packet rather than adding additional erasure-coded packets to lower the rate of the packet-level erasure code.

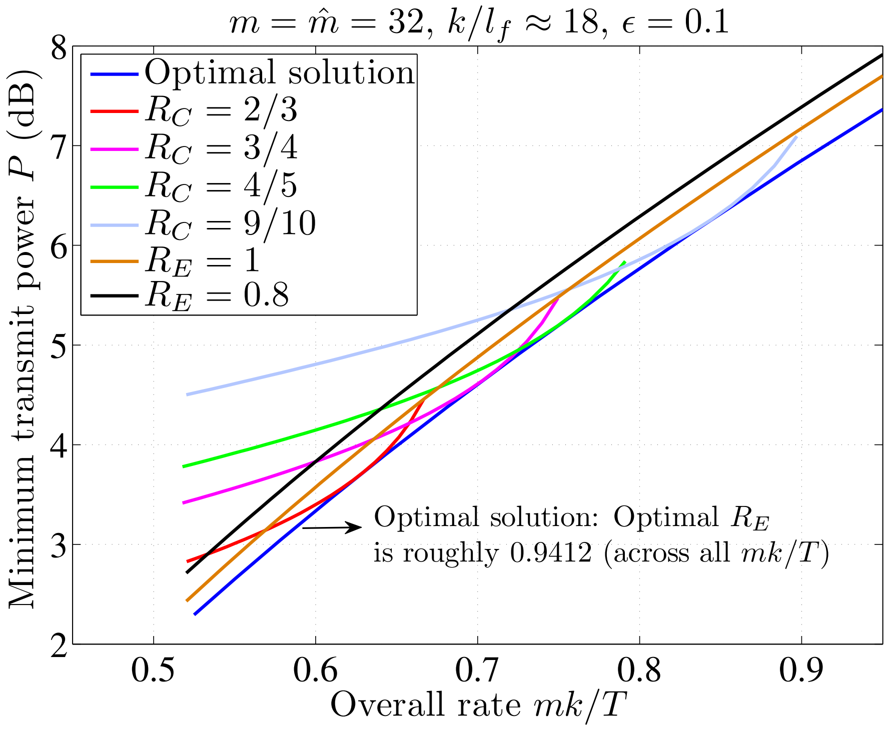

Figure 5.6: Optimal system at each mk/T compared against suboptimal systems with either a fixed channel code rate or a fixed erasure code rate

To view the benefit in using the optimal R_E and R_C values at each overall rate in a system where diversity grows with block-length, we plot in Fig. 5.6 the minimum required power to operate with $q \leq \lambda$ in various fixed systems. We see in the figure that the optimal system obtained by solving the optimization problem at each overall rate can outperform a system with a fixed value of R_C by as much as 2.25 dB. In general, the difference in power levels depends upon the system parameters, mainly m, $\hat{m}$, and $\frac{k}{l_f}$. The optimization routine can determine if a fixed-rate (fixed channel code rate or fixed erasure code rate) solution is good enough instead of using rate-compatibility at both layers of coding in a system that employs incremental redundancy. In this example, the optimal R_E value for the system under consideration in Fig. 5.6 is shown to be roughly constant at around 0.94 in Fig. 5.5.

The presence of diversity growth in the channel has a significant impact on the optimal value of R_E. Fig. 5.7 illustrates how the diversity growth factor (DGF), $\frac{k}{l_f}$, affects how the optimal R_E value changes with overall rate and compares this behavior to that of fading

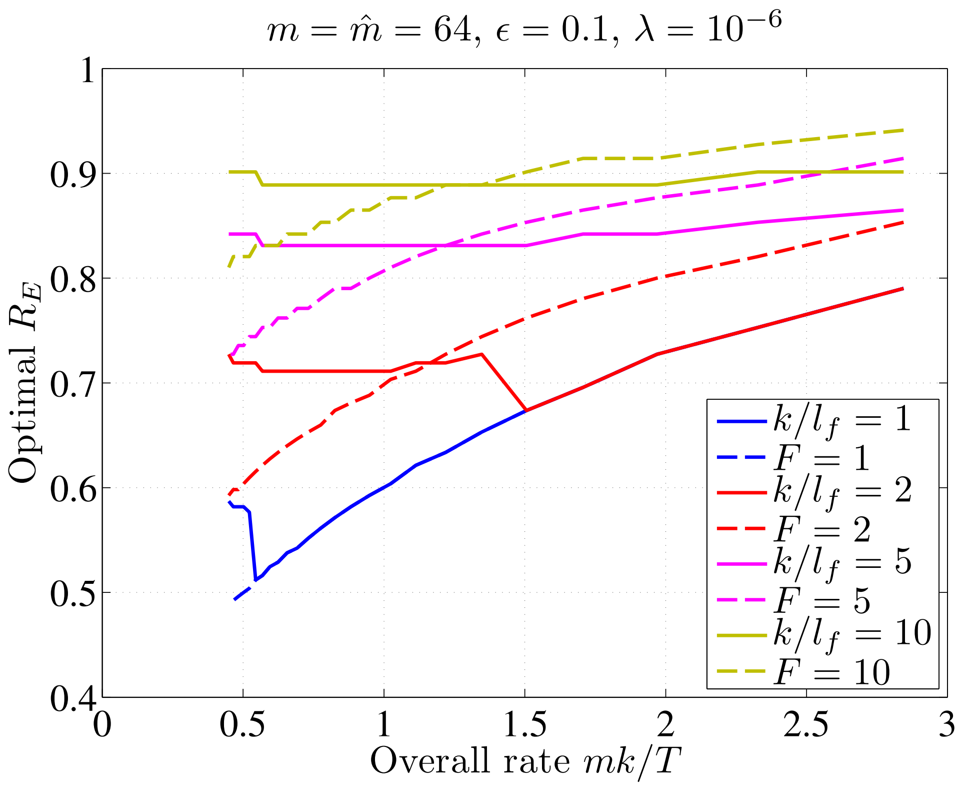

Figure 5.7: Effect of diversity growth factor (DGF), $\frac{k}{l_f}$, on the optimal R_E value: A low DGF leads to an initially decreasing optimal erasure code rate as the overall code rate decreases. A high DGF, on the other hand, indicates that it is better to make only the channel code rateless and keep the erasure code rate roughly constant. Also shown here are the optimal R_E values for fixed-diversity systems with F fades per codeword irrespective of the block-length.

channels without diversity growth. The curves in Fig. 5.7 show the optimal R_E value as a function of the overall rate. Solid curves show this behavior for a block-fading channel with diversity that grows with block-length while dashed curves show this behavior for a block-fading channel with a fixed diversity that does not change with block-length. The optimization problem for the fixed-diversity case, as solved in [CW11], is identical to (5.17) but with $\frac{k}{R_C l_f}$ replaced by the fixed number of fading blocks per codeword.

As observed previously in [CW11], for channels with a fixed amount of diversity irrespective of the block-length, the optimal R_E monotonically decreases as the overall rate decreases, decreasing faster as the overall rate approaches zero. In contrast, channels with diversity that grows with block-length have optimal R_E values for which the rate of decrease

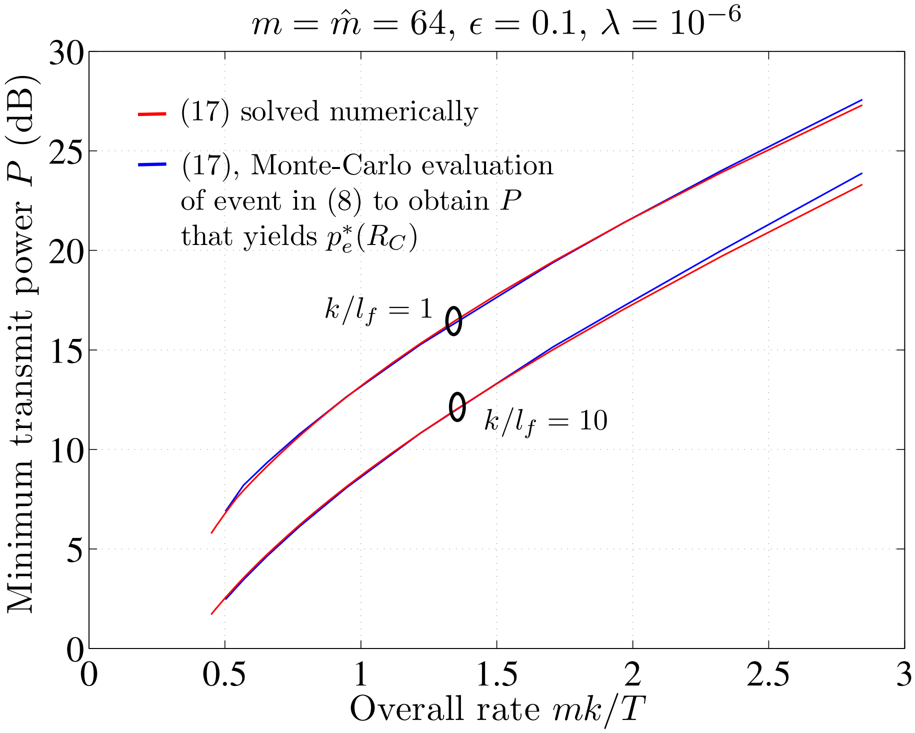

Figure 5.8: Optimal value of the objective function in (5.17): Red line shows the result of the optimization problem in (5.17) solved numerically. Blue line shows the result of performing the bisection search using Monte-Carlo evaluation of (5.8) to obtain the required P that yields $p_e^*(R_C)$.

slows as overall rate approaches zero. Furthermore, as shown in Section 5.4, the optimal erasure code rate will eventually approach 1 (as $m = \hat{m}$) for the systems with diversity that grows with block-length.

Note that there are two places in the curves for channels with diversity growth that display local non-monotonic behavior. One example of this occurs with the $\frac{k}{l_f} = 1$ curve in Fig. 5.7. For an overall rate above 0.55, the optimal R_E for the $\frac{k}{l_f} = 1$ curve exactly tracks the optimal R_E curve for the $F = 1$ fixed-diversity curve. However, as the overall rate decreases to below 0.55, the R_E curve for the PD channel suddenly increases as overall rate decreases. The $\frac{k}{l_f} = 2$ curve in Fig. 5.7 shows similar behavior in which the optimal R_E suddenly increases with decreasing overall rate when the overall rate drops below 1.5.

These points of local irregularity correspond to the change from one approximation to another in the equation for $p_e^*(R_C)$ in (5.17) as the optimal R_C decreases to be below $\frac{k}{l_f}$.

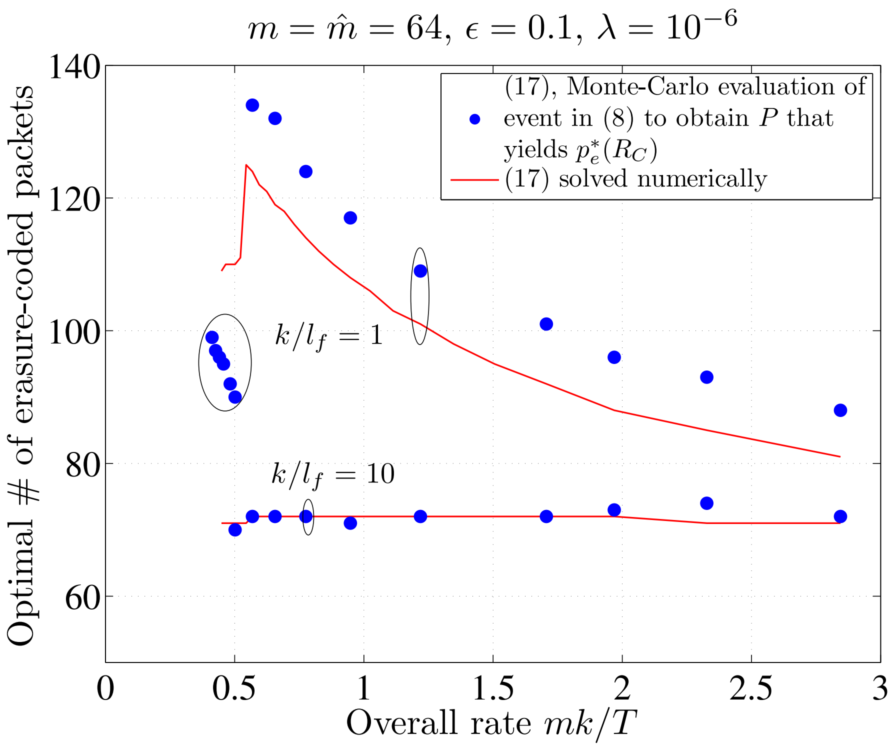

Figure 5.9: Optimal number of erasure-coded packets

However, we have confirmed that this discontinuity is more than an artifact of changing approximation techniques. Fig. 5.8 shows the optimal value of the objective function as obtained by solving the numerical optimization problem in (5.17) and by directly solving the optimization problem without any approximation by using a Monte-Carlo evaluation of p_e in (5.8) to obtain P that yields $p_e^*(R_C)$ through the bisection search.

Furthermore, Fig. 5.9 shows that the Monte-Carlo based optimization that uses no approximations displays the same discontinuity for the $\frac{k}{l_f} = 1$ case. Rather than being a result of changing approximation techniques, this discontinuity is a result of the system changing "phase" from a system with no diversity to a system with two branches of diversity. The abrupt introduction of diversity suddenly adds much greater importance to the physical-layer code that can harvest the newly available diversity. To allow the physical-layer code to benefit from the new diversity, symbols are taken away from the erasure code and given to the physical-layer code, creating the discontinuity seen in the $\frac{k}{l_f} = 1$ curve in Figs. 5.7 and 5.9 for the overall rate near 0.55.

While Fig. 5.9 shows good agreement between the optimal number of erasure-coded packets found by direct Monte-Carlo and by the Gaussian approximation (5.17) for the $\frac{k}{l_f} = 10$ case, we see a gap between the optimal number of erasure-coded packets found by direct Monte-Carlo and by the Gaussian approximation for the $\frac{k}{l_f} = 1$ case. This gap is not unexpected since with only one fade the Gaussian approximation is not quite accurate for computing p_e. However, the flatness of the objective function for large values of T similar to Fig. 5.4 allow the optimal objective functions shown in Fig. 5.8 to match well even though the optimal number of erasure-coded packets found by direct Monte-Carlo and by the Gaussian approximation method do not completely agree in Fig. 5.9.

5.5.2 Very Low Overall Rates

Now, consider what happens to optimal R_E as the overall rate goes to zero (as $T \to \infty$). When the overall rate is very low, the minimum SNR required for reliable transmission is correspondingly small. Thus, at these very low overall rates we use a low-SNR approximation when performing the bisection search to obtain P that yields $p_e^*(R_C)$ in (5.17). Specifically, for a very low value of transmit power P, $\ln(1+\gamma) \approx \gamma$ with very high probability. Hence for the Rayleigh fading channel, we assume that $\mu(P) = \mathbb{E}[\ln(1+\gamma)] \approx \mathbb{E}[\gamma] = P$ and that $\text{Var}(P) = \text{Var}(\ln(1+\gamma)) \approx \text{Var}(\gamma) = P^2$.

For channels with diversity growth, Fig. 5.10 shows curves of optimal R_E for overall rates close to zero. It is seen from the results in Fig. 5.10 that regardless of the value of $\frac{k}{l_f}$, the optimal value of R_E is increasing towards 1 as the overall rate decreases. As shown in Section 5.4, for channels with diversity growth, the optimal R_E is *equal to 1* for any low-enough overall rate for any value of $\frac{k}{l_f}$.

This behavior is different than what was seen in [CW11] for channels with fixed diversity. As shown in Section VI of [CW11], for the fixed-diversity case as overall rate decreases to zero the optimal values of R_E converge to values that are bounded away from zero but can generally take any value in the region $0 < R_E \leq 1$. For the fixed-diversity channels studied

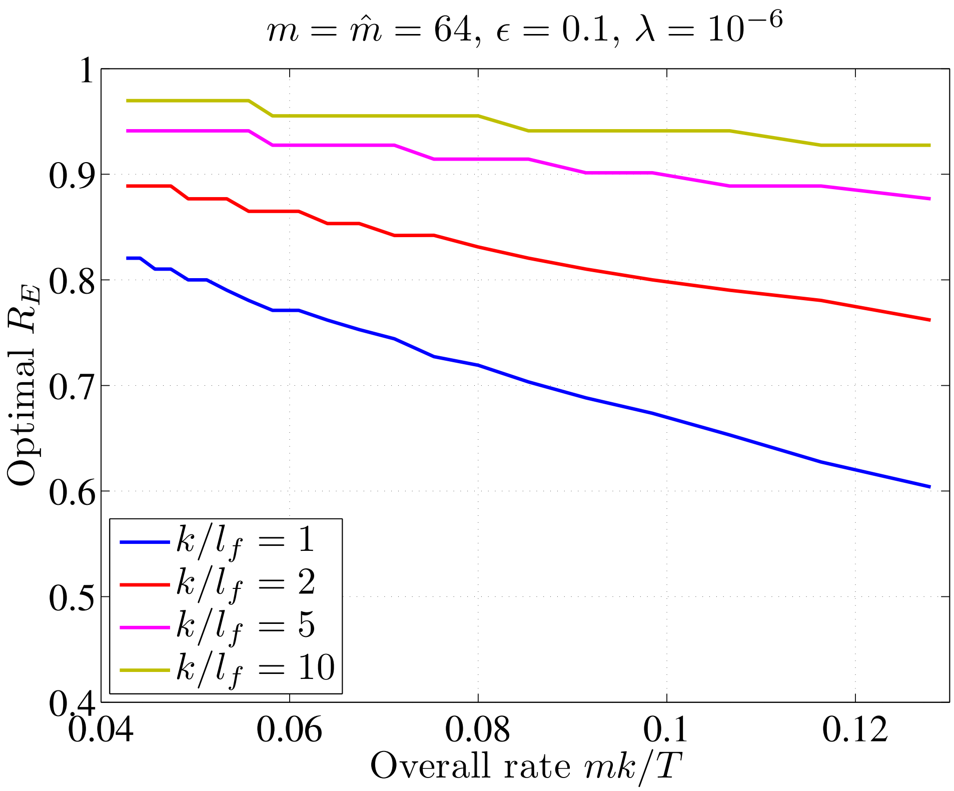

Figure 5.10: Increase of optimal R_E with respect to decreasing overall rates at values of very low overall rate in fading channels with diversity growth

Table 5.1: Asymptotic Values of Optimal R_E for the Fixed-Diversity Systems from Fig. 5.7. The Values Were Computed Using the Approach in Section VI of [CW11].

F	1	2	5	10
Opt. R_E, $T \to \infty$	0.2735	0.4267	0.6275	0.7529

in Fig. 5.7, the asymptotic optimal values of R_E as overall rate goes to zero are provided in Table 5.1.

However, even in [CW11] it was observed in the context of fixed-diversity channels that whenever the number of block fades per channel codeword is large enough, erasure coding becomes insignificant to the system in terms of the benefit it yields and optimal values of R_E are very close to 1. Thus, the low-rate results we present for channels with diversity growth are consistent with the results observed in [CW11].

5.6 A Related Optimization Problem

In this section, we discuss the results of numerically solving a related optimization problem of minimizing the probability of message error given a fixed transmitter power. The motivation to consider this alternative optimization problem is to identify whether the trends observed in the solution to the optimization problem of Section 5.3 also appear under this alternative optimization problem. If the optimal erasure code rate behaves similarly, then the conclusions of our work hold in a much broader context.

The optimization problem is written as

$$\begin{aligned} &\min_{R_C} \quad q \\ &\text{s.t.} \quad R_C > 0 \text{ satisfying } \hat{m} \leq \frac{m}{R_E},\ \frac{m}{R_E} \in \mathbb{Z}. \end{aligned} \tag{5.20}$$

where q, which depends upon the number of erasure-coded packets $R_C T k^{-1}$ that are transmitted and the probability of codeword decoding error p_e, is defined in (5.5). Note that the fixed transmitter power P controls p_e. The solution to this optimization problem is obtained in a relatively straight-forward manner unlike the optimization problem of Section 5.3. For a fixed value of the overall code rate, we compute the value of q for each R_C (hence R_E) that is under consideration and the optimal R_C value is the value that minimizes q. Note that minimizing q, which is the value of the cumulative distribution function $\Phi(x)$ of a standard normal distribution, is equivalent to minimizing the argument of the function.

For our numerical results discussed here, p_e is computed according to (5.14), which is the approximation that is also used in the results related to the optimization problem of Section 5.3. Fig. 5.11 shows the results of numerically solving the optimization problem (5.20). Similar to the results of Section 5.5, a larger value of $\frac{k}{l_f}$ results in a relatively higher optimal erasure code rate across the range of overall code rates considered here. Furthermore, the optimal erasure code rate increases to its maximum possible value (as a function of decreasing overall code rate) beyond a certain value of the overall code rate. This behavior was observed across different power levels and system parameters.

We note that minimizing the message error probability without any lower bound on the

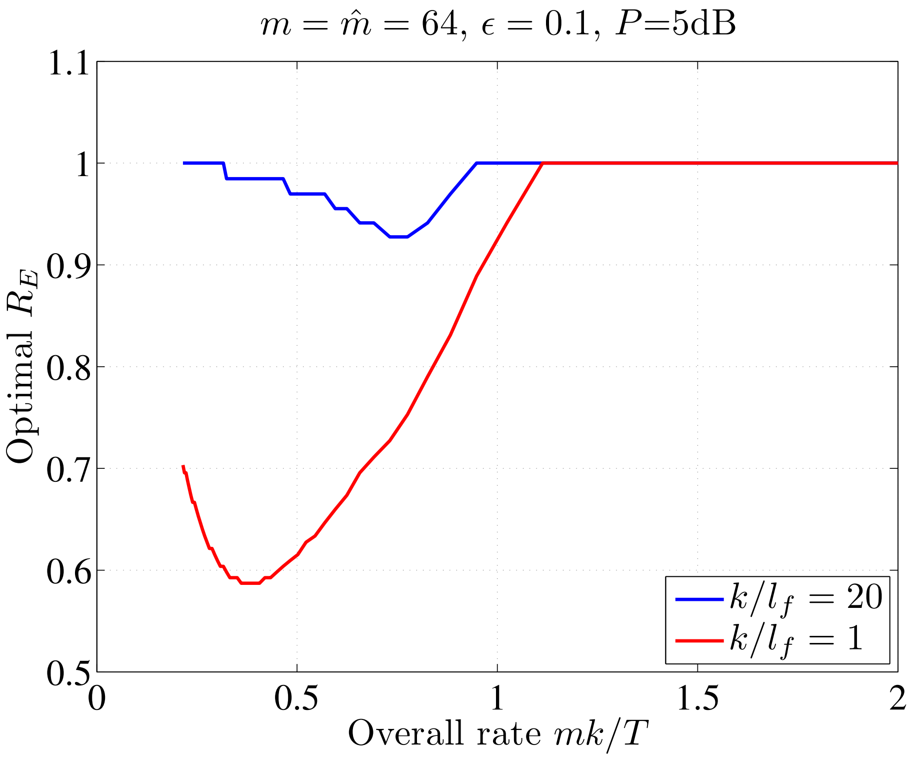

Figure 5.11: Optimal R_E as a function of the overall code rate mk/T for the optimization problem (5.20)

same is not a practical problem to be posed in real systems. This is because real systems only require a certain target message error probability to be achieved. Our results in Fig. 5.11 do not have any such lower bound on the target message error probability. If there is such a lower bound included as a constraint to the optimization problem (5.20), we obtain multiple R_E values that are optimal, which allows for a greater flexibility in system designs.

5.7 Concluding Remarks

This chapter explores the optimal allocation of redundancy between inter-packet (packet-level) erasure coding and intra-packet (physical-layer) channel coding for fading channels where diversity grows with codeword length. This is a practically important scenario for time-varying channels for which a fixed coherence time will lead to a growth in diversity as the number of transmitted symbols increases. As the overall rate goes to zero, the optimal packet-level erasure code rate goes to 1 (when $m = \hat{m}$), indicating that no packet-level erasure coding is necessary in the limit of extremely low rates and the resulting extremely high diversity.

Perhaps a more practical result is that the optimal value of the erasure code rate R_E remains relatively flat for some interesting rate regions, especially when the diversity growth factor $\frac{k}{l_f}$ is sufficiently large. We presented an example demonstrating that for sufficiently large values of $\frac{k}{l_f}$ an optimized system can have an optimal erasure code rate that is a fixed constant. In contrast, the optimal rate R_C of the physical-layer channel code decreases monotonically to zero as the overall rate goes to zero for channels with diversity growth. Thus, when considering systems with incremental redundancy, optimal rate allocation for fading channels with diversity growth requires a physical-layer code that is "rateless".

We recognize that implementation of a "rateless" code for the physical layer presents some practical challenges as compared with "rateless" coding for correction of packet-level erasures. In particular, rateless codes at the physical layer require that received packets that could not be decoded must be stored until additional redundancy is received to facilitate a subsequent decoding attempt. In contrast, with rateless packet-level erasure coding, packets that are not successfully decoded are simply discarded and treated as erasures. However, the point of this chapter is that rateless coding at the physical layer provides benefits not available with rateless coding for packet-level erasures, so that system designers should at least consider whether the benefit might be worth that additional implementation complexity.

5.8 Appendix

5.8.1 Proof that p_e in (5.14) is a Strictly Increasing Function of R_C

Lemma 12. *The Gaussian approximation to the probability of codeword decoding error p_e in* (5.14) *is a strictly increasing function of R_C.*

Proof. First, note that the function is continuous as the expressions for the two cases evaluate to the same value when $R_C = \frac{k}{l_f}$ and the two component functions are continuous. Furthermore, $\Phi\left[\frac{cR_C-\mu(P)}{\sqrt{\mathrm{Var}(P)}}\right]$ is a strictly increasing function in R_C, as is clear from its arguments. We now show that $\Phi\left[\sqrt{\frac{k}{R_C l_f}}\frac{cR_C-\mu(P)}{\sqrt{\mathrm{Var}(P)}}\right]$ is a strictly increasing function in R_C by observing that its derivative is positive for all values $R_C > 0$:

$$\begin{aligned}\frac{d}{dR_C}\Phi\left[\sqrt{\frac{k}{R_C l_f}}\frac{cR_C-\mu(P)}{\sqrt{\mathrm{Var}(P)}}\right] =& \frac{1}{\sqrt{2\pi}}e^{-\frac{\left(\sqrt{\frac{k}{R_C l_f}}\frac{cR_C-\mu(P)}{\sqrt{\mathrm{Var}(P)}}\right)^2}{2}} \\ & \frac{d}{dR_C}\left[\sqrt{\frac{k}{R_C l_f}}\frac{cR_C-\mu(P)}{\sqrt{\mathrm{Var}(P)}}\right] \qquad (5.21)\\ =& c_1\frac{d}{dR_C}\left[\frac{cR_C-\mu(P)}{\sqrt{R_C}}\right] \qquad (5.22)\\ =& c_1\frac{cR_C+\mu(P)}{2R_C^{\frac{3}{2}}}, \qquad (5.23)\end{aligned}$$

where $c_1 > 0$ collects the remaining terms that depend on positive constants $\mu(P)$, $\mathrm{Var}(P)$, k, and l_f, and a varying positive exponential term that has R_C in the exponent. □

5.8.2 Proof of Theorem 11

Proof. We will show that any value of R_E other than $\frac{m}{\hat{m}}$ (corresponding to an R_C value of $\hat{m}k/T$) is suboptimal for any T that is large enough.

Recall that the feasible set of R_E values is such that $\frac{m}{R_E} \in \mathbb{Z}$ and $R_E \in \left\{\frac{m}{\hat{m}}, \frac{m}{\hat{m}+1}, \frac{m}{\hat{m}+2}, \ldots\right\}$. For any fixed R_E, the probability that a codeword cannot be decoded must be $p_e^*(R_C) = p_e^*\left(\frac{mk}{TR_E}\right)$ so that the overall probability of message error is equal to the allowable value λ. This value of $p_e^*\left(\frac{mk}{TR_E}\right)$ is achieved by using the lowest possible value of transmit power P for this fixed value of R_E.

First, let us consider an arbitrary value of T. When R_E is fixed at $R_{E_0} \triangleq \frac{m}{\hat{m}}$, we transmit $\hat{m}$ erasure-coded packets using a channel code for each packet. All $\hat{m}$ channel codewords must be decoded correctly by the channel decoder to recover the message. Let p_{n_0} be the probability of codeword error "needed" by the system to achieve a message error probability of $q = \lambda$ at R_{E_0}. The value of p_{n_0}, using the exact expression rather than the approximation in (5.5), can be obtained as follows:

$$\lambda = 1 - (1 - p_{n_0})^{\hat{m}} \tag{5.24}$$

$$\iff p_{n_0} = 1 - \sqrt[\hat{m}]{1 - \lambda}. \tag{5.25}$$

This value $0 < p_{n_0} < 1$ is the required probability of decoding error of the channel code in order to meet the message error rate target of λ.

Given that $R_E = \frac{m}{\hat{m}}$, $R_C = \frac{\hat{m}k}{T}$. We now set the transmit power, which we call $P_0(T)$, using (5.14) with the assumption in the theorem that $R_C \leq \frac{k}{l_f}$, so that the channel code can "provide" a probability of decoding error $p_{p_0} = p_{n_0}$:

$$p_{p_0} = \Phi \left[\sqrt{\frac{T}{\hat{m} l_f}} \frac{\frac{c\hat{m}k}{T} - \mu(P_0(T))}{\sqrt{\mathrm{Var}(P_0(T))}} \right]. \tag{5.26}$$

Now, we assume that T is large enough that $P_0(T)$, which is the required transmit power at this T and $R_E = \frac{m}{\hat{m}}$, is very low. So we employ the same low-SNR approximation as explained in Section 5.5 that $\mu(P) \approx P$ and the variance $\mathrm{Var}(P) \approx P^2$. Using this low-SNR approximation, we get:

$$p_{p_0} = \Phi \left[\sqrt{\frac{T}{\hat{m} l_f}} \frac{\frac{c\hat{m}k}{T} - P_0(T)}{P_0(T)} \right]. \tag{5.27}$$

By manipulating this expression, we can obtain that

$$P_0(T) = \frac{\sqrt{\frac{T}{\hat{m} l_f}} \frac{c\hat{m}k}{T}}{\left[\Phi^{-1}(p_{p_0}) + \sqrt{\frac{T}{\hat{m} l_f}} \right]}, \tag{5.28}$$

where $0 < p_{p_0} = p_{n_0} < 1$. Looking at the expression for p_{n_0} in (5.25), one can expect λ to be quite low so that p_{n_0} is very close to zero and certainly less than $\frac{1}{2}$. This means that p_{p_0} in (5.27) is less than a half and hence the argument inside the $\Phi(\cdot)$ function is negative. This implies that $P_0(T) > \frac{c\hat{m}k}{T}$ and thus positive when obtained explicitly in (5.28) even when $\Phi^{-1}(p_{p_0}) < 0$ in the denominator of (5.28).

Now, for the same fixed T let us see whether $P_0(T)$ can possibly work for any other value of R_E in the feasible set $R_{E_i} = \frac{m}{\hat{m}+i}, i \geq 1$. $P_0(T)$ is a transmit power level that will work for the system at R_{E_i} if the probability of decoding error provided by the channel code $p_{p_i}(T)$ is less than or equal to the probability of decoding error needed p_{n_i}. Note that $p_{p_i}(T)$ does depend upon T when $i \geq 1$ unlike p_{p_0}, which was set to be equal to p_{n_0} to make sure that the system works at R_{E_0}. Essentially, we obtain $p_{p_i}(T)$ for $i \geq 1$ by evaluating (5.14) at T, $R_C = \frac{(\hat{m}+i)k}{T}$, and power level $P_0(T)$ via the low-SNR approximation. Collecting all of this, the system achieves $q \leq \lambda$ at T, $P_0(T)$, and R_{E_i} if and only if

$$\Delta_i(T) \triangleq p_{n_i} - p_{p_i}(T) \geq 0. \tag{5.29}$$

Lemma 13. *At any value of T large enough, $R_{E_0} = \frac{m}{\hat{m}}$ is uniquely optimal if and only if $\Delta_i(T) < 0$ for all $i \geq 1$. Moreover, the optimal transmit power in this case, for any sufficiently large T, can be approximated by*

$$P^* = P_0(T) = \frac{\sqrt{\frac{T}{\hat{m}l_f}}\frac{c\hat{m}k}{T}}{\left[\Phi^{-1}(p_{p_0}) + \sqrt{\frac{T}{\hat{m}l_f}}\right]}, \tag{5.30}$$

where $p_{p_0} = 1 - \sqrt[\hat{m}]{1-\lambda}$.

Proof. Follows from the arguments so far. □

Continuing with the main proof, at any value R_{E_i}, $0 < p_{n_i} < 1$ is any solution to the following equation:

$$\lambda = 1 - \left[\sum_{j=\hat{m}}^{\frac{m}{R_{E_i}}} \binom{\frac{m}{R_{E_i}}}{k} (1-p_{n_i})^j (p_{n_i})^{\frac{m}{R_{E_i}}-j}\right] \tag{5.31}$$

We now show that $\lim_{T\to\infty} p_{p_i}(T) = 1, \forall i \geq 1$ so that

$$\lim_{T\to\infty} \Delta_i(T) = \lim_{T\to\infty} p_{n_i} - p_{p_i}(T) < 0, \forall i. \tag{5.32}$$

In particular, since we have assumed that the feasible set of R_E values is finite because R_C is in a restricted interval and $\frac{m}{R_E}$ can only assume integer values, there is a large-enough T_{opt} such that $\Delta_i(T) < 0,\ \forall i,\ T \geq T_{\text{opt}}$.

The following proves that $\lim_{T\to\infty} p_{p_i}(T) = 1, \forall i$ (we drop the dependence of $p_{p_i}(T)$ on T to satisfy column spacing):

$$\begin{aligned}
p_{p_i} =& \Phi\left[\sqrt{\frac{T}{(\hat{m}+i)l_f}}\frac{\frac{c(\hat{m}+i)k}{T} - P_0(T)}{P_0(T)}\right] && (5.33)\\
=& \Phi\left[\sqrt{\frac{T}{(\hat{m}+i)l_f}} \cdot \frac{\frac{c(\hat{m}+i)k}{T} - \frac{\sqrt{\frac{T}{\hat{m}l_f}}\frac{c\hat{m}k}{T}}{\left[\Phi^{-1}\left(p_{p_0}\right)+\sqrt{\frac{T}{\hat{m}l_f}}\right]}}{\frac{\sqrt{\frac{T}{\hat{m}l_f}}\frac{c\hat{m}k}{T}}{\left[\Phi^{-1}\left(p_{p_0}\right)+\sqrt{\frac{T}{\hat{m}l_f}}\right]}}\right] && (5.34)\\
=& \Phi\left[\sqrt{\frac{\hat{m}}{\hat{m}+i}} \cdot \frac{(\hat{m}+i)\left[\Phi^{-1}\left(p_{p_0}\right) + \sqrt{\frac{T}{\hat{m}l_f}}\right] - \sqrt{\frac{T}{\hat{m}l_f}}\hat{m}}{\hat{m}}\right], && (5.35)
\end{aligned}$$

which upon further manipulations becomes

$$\begin{aligned}
p_{p_i} =& \Phi\left[\sqrt{\frac{1}{\hat{m}(\hat{m}+i)}}\left\{(\hat{m}+i)\Phi^{-1}\left(p_{p_0}\right) + i\sqrt{\frac{T}{\hat{m}l_f}}\right\}\right] && (5.36)\\
\to& 1 \text{ as } T \to \infty, \forall i && (5.37)
\end{aligned}$$

This completes the proof of the main theorem. □

Acknowledgment

The work that resulted in this chapter was supported by NSF under grant CIF-1618272.

CHAPTER 6

Open Problems

This dissertation discussed aspects of rate-compatibility (Chapters 1, 2, 3, 5) and protograph LDPC codes (Chapter 4). We now discuss open problems and possible future extensions of some of the work in this dissertation.

Chapter 1 described a new design method based on permanent upper bounds on minimum distance to obtain PBRL codes for short block-length applications that require low FERs. A limitation of the work in this chapter is that simulation results focused entirely on the BI-AWGN channel. Future work could explore whether designs maximizing the permanent bound also perform well on the binary symmetric and binary erasure channels with low FER requirements. Similarly, simulation results were limited to full-precision belief propagation decoding. Further work could explore how designs maximizing the permanent bound perform with other decoder implementations. This chapter concentrated on QC PBRL codes designed only according to the newly proposed design method. One could use the method of optimizing the threshold once enough rows have been added according to the newly proposed permanent-bound-based design method.

Chapter 2 discussed several properties of PBRL codes; there are open problems left to be explored. One important question is whether the convergence property for infinite iterations holds in some sense also under finite iterations. Another key question left unsolved is whether one can provide theoretical guarantees as to how sub-optimal is the PBRL extension when compared to the optimal lower-triangular extension.

Chapter 3 described ways to handle rate-compatibility when the incremental symbols are delivered out-of-order. The chapter presented PBRL codes for universal increment ordering. A key open problem is to show whether or not UIO-PBRL codes with random increment ordering can have the same throughput as a PBRL code designed for conventional rate-compatibility under in-order increment arrivals.

Chapter 4 settled the question of the shortest block-length required to achieve a girth of 6 for $(3, L)$ complete-protograph QC-LDPC codes. Our approach in Chapter 4 indicates that additive combinatorics may help in tightening the existing bound on the shortest block-length required to achieve a girth of 8.

Chapter 5 solved the problem of what the optimal rate allocation scheme is for a cross-layer system with packet-level erasure coding and physical-layer channel coding when the fading channel has diversity that grows with block-length. An open problem we identify here is the development of a technique that can directly compute the least possible value of optimal erasure code rate for a specified set of system parameters when the fading channel has diversity growth. The answer to this question would let a system designer know the range of low values of R_E that need not be considered for any value of overall rate.

References

[ASDR11] S. Abu-Surra, D. Divsalar, and W. E. Ryan. Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes. *IEEE Trans. Inf. Theory*, 57(2):858–886, February 2011.

[Big05] Ezio Biglieri. *Coding for Wireless Channels.* Springer US, 2005.

[Bla03] Richard E Blahut. *Algebraic Codes for Data Transmission.* Cambridge Univ. Press, Cambridge, England, 2003.

[BS13] B. K. Butler and P. H. Siegel. Bounds on the minimum distance of punctured quasi-cyclic LDPC codes. *IEEE Trans. Inf. Theory*, 59(7):4584–4597, July 2013.

[BV09] Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge Univ. Press, Cambridge, England, 2009.

[BZW$^+$08] C. R. Berger, Shengli Zhou, Yonggang Wen, P. Willett, and K. Pattipati. Optimizing joint erasure- and error-correction coding for wireless packet transmissions. *IEEE Trans. Wireless Commun.*, 7(11):4586–4595, November 2008.

[CB10] Yu Cao and S. D. Blostein. Cross-layer optimization of rateless coding over wireless fading channels. In *25th Biennial Symp. Commun.*, pages 144–149, May 2010.

[CCG79] J. Cain, G. Clark, and J. Geist. Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding (Corresp.). *IEEE Trans. Inf. Theory*, 25(1):97–100, January 1979.

[CHK09] Tao Cui, Tracey Ho, and Jorg Kliewer. Achievable rate and optimal physical layer rate allocation in interference-free wireless networks. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 2674–2678, June 2009.

[CVDW15] Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel. Protograph-based raptor-like LDPC codes. *IEEE Trans. Commun.*, 63(5):1522–1532, May 2015.

[CW11] T. A. Courtade and R. D. Wesel. Optimal allocation of redundancy between packet-level erasure coding and physical-layer channel coding in fading channels. *IEEE Trans. Commun.*, 59(8):2101–2109, August 2011.

[DDJA09] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews. Capacity-approaching protograph codes. *IEEE J. Sel. Areas Commun.*, 27(6):876–888, August 2009.

[DKO$^+$16] G. Durisi, T. Koch, J. Östman, Y. Polyanskiy, and W. Yang. Short-packet communications over multiple-antenna rayleigh-fading channels. *IEEE Trans. Commun.*, 64(2):618–629, February 2016.

[Dol05] S. Dolinar. A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1627–1631, September 2005.

[DPT$^+$02] Changyan Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inf. Theory*, 48(6):1570–1579, June 2002.

[EKHB09] M. El-Khamy, J. Hou, and N. Bhushan. Design of rate-compatible structured LDPC codes for hybrid ARQ applications. *IEEE J. Sel. Areas Commun.*, 27(6):965–973, August 2009.

[ES06] O. Etesami and A. Shokrollahi. Raptor codes on binary memoryless symmetric channels. *IEEE Trans. Inf. Theory*, 52(5):2033–2051, May 2006.

[Fan00] John L. Fan. Array codes as low-density parity-check codes. In *Proc. 2nd Int. Symp. Turbo Codes & Related Topics*, pages 543–546, Brest, September 2000.

[Fos04] M. P. C. Fossorier. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. *IEEE Trans. Inf. Theory*, 50(8):1788–1793, August 2004.

[Gal63] Robert Gray Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, USA, 1963.

[GHW$^+$12] Zheng Guo, Jie Huang, Bing Wang, Shengli Zhou, Jun-Hong Cui, and P. Willett. A practical joint network-channel coding scheme for reliable communication in wireless networks. *IEEE Trans. Wireless Commun.*, 11(6):2084–2094, June 2012.

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, New York, 2005.

[Hag88] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Trans. Commun.*, 36(4):389–400, April 1988.

[HEA05] Xiao-Yu Hu, E. Eleftheriou, and D.-M. Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inf. Theory*, 51(1):386–398, January 2005.

[HKKM06] Jeongseok Ha, Jaehong Kim, D. Klinc, and S. W. McLaughlin. Rate-compatible punctured low-density parity-check codes with short block lengths. *IEEE Trans. Inf. Theory*, 52(2):728–738, February 2006.

[HKM04] Jeongseok Ha, Jaehong Kim, and S. W. McLaughlin. Rate-compatible puncturing of low-density parity-check codes. *IEEE Trans. Inf. Theory*, 50(11):2824–2836, November 2004.

[HNK$^+$06] Manabu Hagiwara, Koji Nuida, Takashi Kitagawa, Marc Fossorier, and Hideki Imai. On the minimal length of quasi cyclic LDPC codes with girth greater than or equal to 6. In *Proc. IEEE Int. Symp. Inform. Theory Applicat.*, October 2006. CD-ROM.

[KB13] M. Karimi and A. H. Banihashemi. On the girth of quasi-cyclic protograph LDPC codes. *IEEE Trans. Inf. Theory*, 59(7):4542–4552, July 2013.

[KCY13] Kyung-Joong Kim, Jin-Ho Chung, and Kyeongcheol Yang. Bounds on the size of parity-check matrices for quasi-cyclic low-density parity-check codes. *IEEE Trans. Inf. Theory*, 59(11):7288–7298, November 2013.

[KHKC11] C. Koller, M. Haenggi, J. Kliewer, and D. J. Costello. On the optimal block length for joint channel and network coding. In *Proc. IEEE Inform. Theory Workshop*, pages 528–532, October 2011.

[KHKC14] C. Koller, M. Haenggi, J. Kliewer, and Jr. Costello, D. J. Joint design of channel and network coding for star networks connected by binary symmetric channels. *IEEE Trans. Commun.*, 62(1):158–169, January 2014.

[KRM09] J. Kim, A. Ramamoorthy, and S. W. Mclaughlin. The design of efficiently-encodable rate-compatible LDPC codes. *IEEE Trans. Commun.*, 57(2):365–375, February 2009.

[LC07] Gianluigi Liva and M. Chiani. Protograph LDPC codes design based on EXIT analysis. In *Proc. IEEE Global Telecommun. Conf.*, pages 3250–3254, November 2007.

[LGSW07] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Trans. Broadcast.*, 53(1):235–246, March 2007.

[LMSS01] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, and Daniel A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Inf. Theory*, 47(2):569–584, February 2001.

[LXXL11] D. Lin, M. Xiao, Y. Xiao, and S. Li. Efficient packet combining based on packet-level coding. *Electron. Lett.*, 47(7):444–445, March 2011.

[MN08] J. Matousek and J. Nesetril. *Invitation to Discrete Mathematics*. OUP Oxford, 2008.

[MSC14] D. G. M. Mitchell, R. Smarandache, and D. J. Costello. Quasi-cyclic LDPC codes based on pre-lifted protographs. *IEEE Trans. Inf. Theory*, 60(10):5856–5874, October 2014.

[NN13] T. V. Nguyen and A. Nosratinia. Rate-compatible short-length protograph LDPC codes. *IEEE Commun. Lett.*, 17(5):948–951, May 2013.

[NND12] T. V. Nguyen, A. Nosratinia, and D. Divsalar. The design of rate-compatible protograph LDPC codes. *IEEE Trans. Commun.*, 60(10):2841–2850, October 2012.

[NS97] K. R. Narayanan and G. L. Stüber. A novel ARQ technique using the turbo coding principle. *IEEE Commun. Lett.*, 1(2):49–51, March 1997.

[Pai47] L. J. Paige. A note on finite abelian groups. *Bulletin of the American Mathematical Society*, 53(6):590–593, June 1947.

[PPV10] Y. Polyanskiy, H. V. Poor, and S. Verdu. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory*, 56(5):2307–2359, May 2010.

[RDVW14] S. V. S. Ranganathan, D. Divsalar, K. Vakilinia, and R. D. Wesel. Design of high-rate irregular non-binary LDPC codes using algorithmic stopping-set cancellation. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 711–715, June 2014.

[RDW15a] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel. On the girth of (3,L) quasi-cyclic LDPC codes based on complete protographs. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 431–435, June 2015.

[RDW15b] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel. On the girth of (3,L) quasi-cyclic LDPC codes based on complete protographs. April 2015. arXiv:1504.04975v1 [cs.IT].

[RDW17] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel. Design of improved quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1207–1211, June 2017.

[RDW18] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel. Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths. 2018. Under revision, IEEE Trans. Inf. Theory.

[Ric03] T. J. Richardson. Error floors of LDPC codes. In *Proc. 41st Annu. Allerton Conf. Commun., Control, and Computing*, volume 5, pages 3125–3129, October 2003.

[RL09] William Ryan and Shu Lin. *Channel Codes Classical and Modern.* Cambridge Univ. Press, Cambridge, England, 2009.

[RMW17] S. V. S. Ranganathan, T. Mu, and R. D. Wesel. Allocating redundancy between erasure coding and channel coding when fading channel diversity grows with codeword length. *IEEE Trans. Commun.*, 65(8):3226–3237, August 2017.

[RSU01] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inf. Theory*, 47(2):619–637, February 2001.

[RU01] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inf. Theory*, 47(2):599–618, February 2001.

[RU08] Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory.* Cambridge University Press, 2008.

[RVDW16] S. V. S. Ranganathan, K. Vakilinia, D. Divsalar, and R. D. Wesel. Universal rate-compatible LDPC code families for any increment ordering. In *Proc. 9th Int. Symp. Turbo Codes & Iterative Inf. Processing (ISTC)*, pages 101–105, September 2016.

[RWD18] S. V. S. Ranganathan, R. D. Wesel, and D. Divsalar. Linear rate-compatible codes with degree-1 extending variable nodes under iterative decoding. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1166–1170, June 2018.

[Rys63] H. J. Ryser. Combinatorial mathematics, Carus Mathematical Monographs. 1963.

[Sho06] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, June 2006.

[Slo] N. J. A. Sloane. The on-line encyclopedia of integer sequences. http://oeis.org/A003111. Number of Complete Mappings of The Cyclic Group Z_{2n+1}.

[Sun11] Fan Sun. Two-layer coding rate optimization in relay-aided systems. In *IEEE Veh. Technology Conf. (VTC Fall)*, pages 1–5, September 2011.

[SV12] R. Smarandache and P. O. Vontobel. Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds. *IEEE Trans. Inf. Theory*, 58(2):585–607, February 2012.

[Tan81] R. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory*, 27(5):533–547, September 1981.

[Tec15] Technical Specification Group Services and System Aspects. 3GPP TS 23.246 V13.3.0. Multimedia broadcast/multicast service (mbms) - architecture and functional description, 3rd Generation Partnership Project, December 2015.

[Tec17] Technical Specification Group Radio Access Network; NR. 3GPP TS 38.212 V2.0.0 (2017-12). Multiplexing and channel coding (release 15), 3rd Generation Partnership Project, December 2017.

[Tho03] Jeremy Thorpe. Low-density parity-check (LDPC) codes constructed from protographs. IPN-PR 42-154, JPL, August 2003.

[TJVW03] Tao Tian, C. Jones, J. D. Villasenor, and R. D. Wesel. Construction of irregular LDPC codes with low error floors. In *Proc. IEEE Int. Conf. Commun.*, volume 5, pages 3125–3129, May 2003.

[TJVW04] Tao Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel. Selective avoidance of cycles in irregular LDPC code construction. *IEEE Trans. Commun.*, 52(8):1242–1247, August 2004.

[VDW15] K. Vakilinia, D. Divsalar, and R. D. Wesel. Protograph-based raptor-like LDPC codes for the binary erasure channel. In *Inform. Theory and Applicat. Workshop*, pages 240–246, February 2015.

[VF09] B. N. Vellambi and F. Fekri. Finite-length rate-compatible LDPC codes: a novel puncturing scheme. *IEEE Trans. Commun.*, 57(2):297–301, February 2009.

[VM05] M. Vehkapera and M. Medard. A throughput-delay trade-off in packetized systems with erasures. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1858–1862, September 2005.

[Von13] P. O. Vontobel. The Bethe permanent of a nonnegative matrix. *IEEE Trans. Inf. Theory*, 59(3):1866–1901, March 2013.

[WRW17] H. Wang, S. V. S. Ranganathan, and R. D. Wesel. Approaching capacity using incremental redundancy without feedback. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 161–165, June 2017.

[WVR$^+$16] R. D. Wesel, K. Vakilinia, S. V. S. Ranganathan, T. Mu, and D. Divsalar. Resource-aware incremental redundancy in feedback and broadcast. In *Int. Zurich Seminar Commun. (IZS)*, pages 63–67, March 2016.

[WYD08] Yige Wang, J. S. Yedidia, and S. C. Draper. Construction of high-girth QC-LDPC codes. In *Proc. 5th Int. Symp. Turbo Codes & Related Topics*, pages 180–185, September 2008.

[XMA11] M. Xiao, M. Medard, and T. Aulin. Cross-layer design of rateless random network codes for delay optimization. *IEEE Trans. Commun.*, 59(12):3311–3322, December 2011.

[YB04] M. R. Yazdani and A. H. Banihashemi. On construction of rate-compatible low-density parity-check codes. *IEEE Commun. Lett.*, 8(3):159–161, March 2004.

[Yil11] A. O. Yilmaz. Calculating outage probability of block fading channels based on moment generating functions. *IEEE Trans. Commun.*, 59(11):2945–2950, November 2011.

[ZDN$^+$06] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright. Gen03-6: Investigation of error floors of structured low-density parity-check codes by hardware emulation. In *IEEE Globecom 2006*, pages 1–6, November 2006.

[ZM16] H. Zeineddine and M. M. Mansour. Inter-frame coding for broadcast communication. *IEEE J. Sel. Areas Commun.*, 34(2):437–452, February 2016.