UNIVERSITY OF CALIFORNIA

Los Angeles

Theory and Practice of Non-Binary Graph-Based Codes: A Combinatorial View

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Behzad Amiri

2015

ABSTRACT OF THE DISSERTATION

Theory and Practice of Non-Binary Graph-Based Codes: A Combinatorial View

by

Behzad Amiri

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Lara Dolecek, Chair

We are undergoing a revolution in data. The ever-growing amount of information in our world has created an unprecedented demand for ultra-reliable, affordable, and resource-efficient data storage systems. Error-correcting codes, as a critical component of any memory device, will play a crucial role in the future of data storage.

One particular class of error-correcting codes, known as graph-based codes, has drawn significant attention in both academia and in industry. Graph-based codes offer superior performance compared to traditional algebraic codes. Recently, it has been shown that non-binary graph-based codes, which operate over finite fields rather than binary alphabets, outperform their binary counterparts and exhibit outstanding overall performance. For this reason, these codes are particularly suitable for emerging data storage systems.

In this dissertation, we present a comprehensive combinatorial analysis of non-binary graph-based codes. We perform both finite-length and asymptotic analyses for these codes, providing a systematic framework to evaluate and optimize various families of non-binary graph-based codes.

In the finite-length case, we provide a mathematical characterization of the error floor problem, including a general definition of absorbing sets over non-binary alpha-

bets. We consider several structured low-density parity-check (LDPC) codes, including quasi-cyclic and spatially-coupled codes, as well as unstructured LDPC codes. We offer design guidelines for non-binary LDPC codes with outstanding performance in extremely low error-rate regimes; making them excellent candidates for data storage applications.

In the asymptotic case, we provide a novel toolbox for the evaluation of families of non-binary graph-based codes. By utilizing insights from graph theory and combinatorics, we establish enumerators for a general family of graph-based codes which are constructed based on protographs. We provide asymptotic distributions of codewords and trapping sets for the family of protograph-based codes. Furthermore, we present an asymptotic enumeration of binary and non-binary elementary absorbing sets for regular code ensembles.

The contributions of this dissertation can potentially impact a broad range of data storage and communication technologies that require excellent performance in high-reliability regimes.

The dissertation of Behzad Amiri is approved.

Mario Gerla

Richard D. Wesel

Alan J. Laub

Lara Dolecek, Committee Chair

University of California, Los Angeles

2015

To my parents, Jalil and Fereshteh

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This dissertation would not be possible without the guidance, support, and companionship of many individuals, to whom I give my deepest thanks and appreciation.

I would like to first thank my advisor, Professor Lara Dolecek, for her guidance and support over the years. I am truly grateful for her help throughout my Ph.D. research. Professor Dolecek is always a great source of insight, new ideas and motivation. Many of the ideas in this dissertation came from my discussions with her. Her emphasis on critical thinking and improved written and verbal communication skills has greatly influenced my graduate studies, and will continue to help me in my future career.

I would like to thank Professor Alan Laub, Professor Richard Wesel, and Professor Mario Gerla for being on my Ph.D. qualifying exam and dissertation committees, as well as reviewing this dissertation. I am very grateful for their time and their guidance.

I am grateful to all my collaborators for their help and insight through these years. It was a great learning experience working with Professor Jörg Kliewer, Professor Dariush Divsalar, and Professor Dejan Markovic. I was also fortunate to work with Arturo Flores, Yuta Toriyama, Yizeng Sun, Dr. Jiadong Wang and Dr. Shayan Garani Srinivasa. The members of the LORIS research lab at UCLA, Frederic Sala, Ahmed Hareedy, Clayton Schoeny, Amirhossein Resisizadeh, Nicolas Bitouze, Dr. Sean Huang, Dr. Ryan Gabrys, Dr. Sadegh Tabatabaei Yazdi, and Dr. Yao Li have been great companions over the past several years. They provided a great friendly and interactive environment at the LORIS lab, which made my Ph.D. experience much more fun.

I must also thank my dear friends whom I shared so many unforgettable memories with, Sina Basir-Kazeruni, Mahdi Ashktorab, Sina Kalbasi, Ehsan Ebrahimzadeh, Ahmad Ghadiri, Koosha Marashi, Mohammadreza Saba, Farid Hendi, Arman Ansari and Farid Yaghini. Spending time with all of you was joyful and fun, and it thought me a lot of invaluable life lessons.

I would like to thank my parents for their constant love, support, and faith in me

throughout my life. My parents raised me up and gave me everything I needed. They always put me before themselves. Without them, I would not have accomplished my goals. My special thanks to my brothers, Babak and Bahador, whose constant and unwavering support have kept me going during the ups and downs of my life. Their love and encouragement significantly helped me during my graduate school career, which included some of the most difficult challenges I have had yet to face.

Last but certainly not least, I cannot find words to express my gratitude for my girlfriend, Ghazaleh. It is no overstatement that none of this would be possible without her relentless support, unconditional and unmeasurable love. She is always supportive no matter what choices I make, many times at her own sacrifice. She is my best friend, and my best supporter.

This work was supported in part by the National Science Foundation (NSF) and the Advanced Storage Technology Consortium (ASTC).

VITA

2010	B.S. in Electrical Engineering Isfahan University of Technology
2009 - 2010	Research Assistant Isfahan University of Technology
2011 - 2015	Research Assistant Electrical Engineering Department University of California, Los Angeles
2011 - 2012	Research Assistant Physics Department University of California, Los Angeles
2012	M.S. in Electrical Engineering University of California, Los Angeles
2013 - 2015	Teaching Assistant Electrical Engineering Department University of California, Los Angeles

PUBLICATIONS

B. Amiri, S. Garani Srinivasa, and L. Dolecek, "Quantization, Absorbing Regions and Practical Message Passing Decoders," In Proceedings of the IEEE Asilomar Conference on Signals, Systems and Computers, Monterey, CA, November 2012.

B. Amiri, C. W. Lin, and L. Dolecek, “Asymptotic Distribution of Absorbing Sets and Fully Absorbing Sets for Regular Sparse Code Ensembles," IEEE Transactions on Communications, vol. 61, no. 2, pp. 455 - 464, February 2013.

B. Amiri, J. Kliewer, and L. Dolecek, “Analysis and Enumeration of Absorbing Sets for Non-Binary Graph-Based Codes,” In Proceedings of the IEEE International Symposium on Information Theory, Istanbul, Turkey, July 2013.

B. Amiri, J. Kliewer, and L. Dolecek, “Analysis and Enumeration of Absorbing Sets for Non-Binary Graph-Based Codes,” IEEE Transactions on Communications, vol. 62, no. 2, pp. 398 - 409, February 2014.

L. Dolecek, D. Divsalar, Y. Sun, and **B. Amiri**, “Non-Binary Protograph-Based LDPC Codes: Enumerators, Analysis, and Designs,” IEEE Transactions on Information Theory, vol. 60, no. 7, pp. 3913 - 3941, July 2014.

Y. Toriyama, **B. Amiri**, L. Dolecek, and D. Markovic, “Field-Order Based Hardware Cost Analysis of Non-Binary LDPC Decoders,” In Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers, Monterey, CA, November 2014.

B. Amiri, A. Flores, and L. Dolecek, “Design of Non-binary Quasi-Cyclic LDPC Codes by Absorbing Set Removal,” In Proceedings of the IEEE Information Theory Workshop, Hobart, TAS, Australia, November 2014.

Y. Toriyama, **B. Amiri**, L. Dolecek, and D. Markovic, “Logarithmic Quantization Scheme for Reduced Hardware Cost and Improved Error Floor in Non-Binary LDPC Codes,” In Proceedings of the IEEE Globecom, Austin, TX, December 2014.

B. Amiri, A. Reisizadeh, J. Kliewer, and L. Dolecek, "Optimized Array-Based Spatially-Coupled LDPC Codes: An Absorbing Set Approach," In Proceedings of the IEEE International Symposium on Information Theory, Hong Kong, June 2015.

B. Amiri, A. Reisizadeh, J. Kliewer, and L. Dolecek, "Optimized Design of Finite-Length Spatially-Coupled Codes: An Absorbing Set-Based Analysis," IEEE Journal on Selected Areas in Communications, 2015, submitted.

A. Hareedy, **B. Amiri**, and L. Dolecek, "Non-Binary LDPC Code Optimization for Partial-Response Channels," IEEE Globecom, 2015, submitted.

CHAPTER 1

Introduction

We are facing exponential growth in the volume of digital data. This growth is predominantly driven by ever-growing demand due to technological advancements in entertainment, communication, safety and security, scientific research, and numerous other fields. As a result, the volume of data is predicted to increase from 2 zettabytes[1] in 2009 to around 44 zettabytes in 2020, as shown in Figure 1.1.

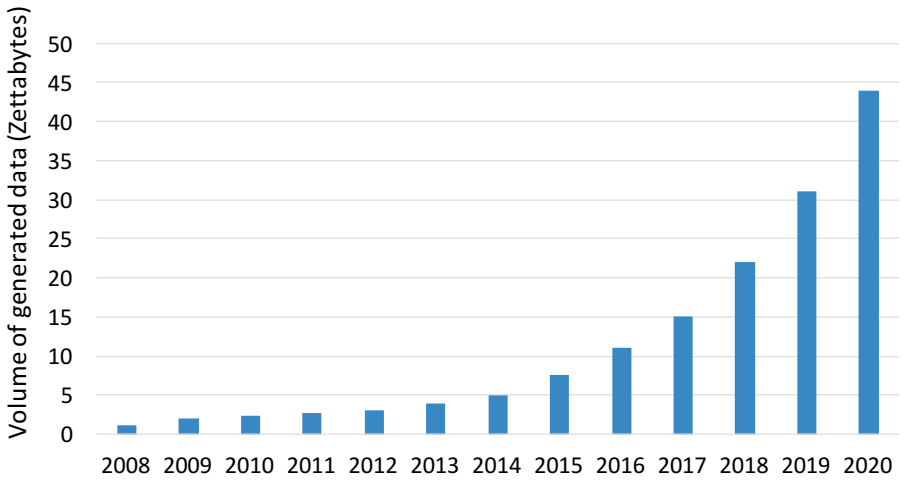


Figure 1.1: Annual volume of data from 2009 to 2020.

The majority of this enormous volume of generated data needs to be stored, resulting in the construction of multi-billion dollar data centers worldwide. For example, a Utah data center, with a capacity of 10 exabytes[2], was built in 2014 at a cost of 1.5 billion dollars. Advanced solutions are essential to the creation of more reliable and less

[1] 1 zettabyte is equal to 10^{21} bytes.
[2] 1 exabyte is equal to 10^{18} bytes.

expensive data storage systems. High-performance error-correcting codes are one such strategy. Employing high-performance error-correcting codes results allows for more reliable storage of data, which leads to improved device lifetimes and thus a significant reduction in cost of storage systems.

Problems related to error-correction over noisy channels were first studied by Claude Shannon in the 1940's. In 1948, Shannon published a landmark paper pioneering the concept of reliable data transmission over noisy channels [1]. He proposed a system consisting of an encoder, a channel, and a decoder. Shannon's work revealed that beyond a certain limiting data rate, called the capacity of the channel, reliable transmission is impossible and that, for data rates below the channel capacity, information can be transmitted with error probability approaching zero. The channel coding theorem [1] proves that there always exists a coding scheme with the property that the decoding error approaches zero exponentially fast by increasing the blocklength of the code. Consequently, there has been a vast amount of research focused on designing codes that are easy to encode and decode in practice and that can also operate close to the channel capacity.

Linear codes are a class of channel codes whose codewords form a linear vector space over a finite field. Linear codes were first proposed by Elias [2] in 1955, and were shown to approach the capacity of discrete memoryless channels. For a linear code, the parity-check matrix is defined as a matrix whose rows span the null-space of the code. Thus, linear codes can be defined in terms of their parity-check matrices as well. Encoding of linear codes can be done by the multiplication of the information vector with a generator matrix. Consequently, linear codes are amenable to low complexity encoding. Moreover, many algebraic linear codes have polynomial time decoding algorithms, such as Hamming codes [3], BCH codes [4], and Reed-Solomon codes [5]. However, none of these codes can reach the capacity of additive white Gaussian noise (AWGN) channels.

The discovery of capacity-approaching turbo codes [6] along with the rediscov-

ery of low-density parity-check (LDPC) codes [7], [8] ushered in a new era of coding research. This period has been marked by research on so-called graph-based code constructions. In particular, the excellent performance of LDPC codes has resulted in their growing use in many applications. LDPC codes reach outstanding near-capacity performance with acceptable encoding/decoding complexity [9]. Therefore, the analysis and design of LDPC codes and their decoders has drawn a lot of significant attention.

Recently, it has been shown that non-binary LDPC codes offer performance improvements compared to their binary counterparts [10]. As a result, several works have been devoted to the construction of various non-binary LDPC codes [11–13]. In addition to code design, the development of low-complexity message-passing decoders with performance close to maximum-likelihood decoding has also garnere research attention [14, 15].

When relying on message-passing decoders, it is known that certain non-codewords vie with codewords to be the output of the decoder. The presence of these non-codeword objects can significantly undermine the performance of iteratively decoded graph-based codes and may even result in an undesirable error floor. Given the significance of the error floor behavior for the finite block-length performance of coding schemes, extensive recent works have been devoted to studying this phenomenon for the binary case, such as [16, 17]. In contrast, for the non-binary case much less is known about how non-codeword objects and specific substructures in the Tanner graph affect the error floor performance [18, 19].

This dissertation sheds light on various challenges facing the utilization of non-binary graph-based codes. We present an analysis of and designs for non-binary graph-based codes . We begin by characterizing the error floor problem by combinatorially defining and analyzing non-binary absorbing sets for codes over non-binary alphabets. We propose an efficient mathematical finite-length code design for structured non-binary quasi-cyclic and spatially-coupled LDPC codes.

Using advanced combinatorics and graph theory, we present an asymptotic enu-

meration of non-binary elementary absorbing sets in regular LDPC code ensembles. Moreover, we provide a comprehensive study of non-binary LDPC codes built from protographs, including both finite-length and asymptotic codeword and trapping set enumerators.

1.1 Summary of Contributions

We briefly outline the contributions of each chapter below.

Chapter 2 Contributions

In this chapter, we provide preliminary definitions and concepts which are necessary for the remainder of this work. We begin with an overview of the Tanner graph representation of linear block codes. Next, two decoding methods of graph-based codes, maximum likelihood decoding and iterative message-passing decoding, are presented. Objects of interest in binary codes, including binary trapping sets and binary absorbing sets are then defined. These definitions are later used as the basis of corresponding objects in non-binary regimes. We conclude this chapter by providing constructions of binary array-based, non-binary quasi-cyclic, and non-binary protograph-based LDPC codes. These families of codes will be considered in our analysis throughout this dissertation.

Chapter 3 Contributions

The main contributions of this chapter are as follows:

- to define and analyze non-binary absorbing sets for codes defined over finite fields,
- to show that the proposed definition for non-binary absorbing sets is valid over different decoders and code constructions,
- to highlight the difference between binary absorbing sets and non-binary absorbing sets, and

- to propose an efficient algorithm based on our classification of absorbing sets in order to design non-binary quasi-cyclic and unstructured LDPC codes.

Chapter 4 Contributions

This chapter has two main directions:

First, we establish the asymptotic analysis of absorbing sets for regular LDPC code ensembles, and, for the first time, we quantify the difference between the asymptotic distributions of absorbing sets and trapping sets. When the normalized logarithmic asymptotic absorbing set distribution (to be defined later) is almost identical to the normalized logarithmic asymptotic trapping set distribution, most trapping sets are in fact absorbing sets, and can represent the decoding errors accurately. On the other hand, when the discrepancy between the trapping set and absorbing set distributions is large, many trapping sets are not stable under finite-precision iterative decoding. Such quantitative observations can be useful in refining algorithms that efficiently search for absorbing sets (or relevant trapping sets).

Second, we focus our attention on the characterization of non-binary LDPC codes built out of protographs. In particular, we consider novel non-binary code constructions that are based on repeating the nodes and permuting the edges as in the binary case, but are also equipped with the added freedom of choosing the non-binary edge weights. We generalize existing definitions and techniques from the binary to the non-binary domains. We also provide an ensemble performance evaluation of the resulting non-binary protograph-based (NBPB) codes through the explicit computation of codeword enumerators and key non-codeword enumerators.

Chapter 5 Contributions

In this chapter, we present a complete characterization of absorbing sets for binary and non-binary array-based spatially-coupled (AB-SC) codes. Our contribution in this chapter is multifold:

- we introduce an analytical approach to find the exact number of absorbing sets in non-binary spatially-coupled codes: the original counting problem is mapped

to a problem of finding the number of integer points within an area in two-dimensional space,

- we find the optimal cutting vector for AB-SC codes with an arbitrary circulant size: analytical and experimental results reveal that the choice of the cutting vector significantly affects the error floor performance of binary and non-binary spatially-coupled codes, and

- we calculate the average number of absorbing sets in non-binary spatially-coupled codes constructed by uninformed (random) assignment of edge weights on top of a binary spatially-coupled code. This result reveals that the average number of absorbing sets in non-binary spatially-coupled codes is significantly lower than the average number of absorbing sets in binary spatially-coupled codes. This explains the superior error floor performance of non-binary spatially-coupled codes compared to their binary counterparts.

CHAPTER 2

Preliminaries of Graph-Based Codes

Most of this dissertation deals with the analysis and design of different families of graph-based codes, and hence we devote this chapter to some basic background material concerning these codes.

2.1 Graphical Representation of Linear Codes

In 1981, Tanner proposed a graphical representation of a parity-check matrix of a linear code [20]. Pearl invented a belief propagation (BP) algorithm as a message passing algorithm to perform inference over Bayesian networks [21]. Now, BP is used in many signal processing, digital communication, and artificial intelligence algorithms such as Viterbi algorithm, turbo decoding, Kalman filtering, etc. One of the important applications of BP is in practical decoding of LDPC codes. The performance of a BP decoder depends on the Tanner graph that characterizes the LDPC code. On a Tanner graph with no cycles, the BP decoder performs the same as the maximum-likelihood decoder [22] which is the optimal decoder. The existence of short cycles in the Tanner graph affects the performance of BP decoder and can remarkably deteriorate the error rate of BP decoders. Consequently, the Tanner graph representation of a code has an important role in the decoding of an LDPC code.

Definition 1. *The Tanner graph G of a code with the $(n-k)\times n$ party-check matrix H is a bipartite graph such that each one of n bit nodes corresponds to one of the n columns of the matrix H. Likewise, each one of $n-k$ check nodes corresponds to one*

of the $n-k$ rows of the matrix H. There exists an edge between bit node j and check node i if and only if the entry in ith row and jth column of matrix H, h_{ij} , is nonzero. Moreover, in the case of non-binary linear codes, each edge is labeled according to the corresponding entry in the parity-check matrix.

Figure 2.1 shows the Tanner graph of a length-7, Hamming code associated with the parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

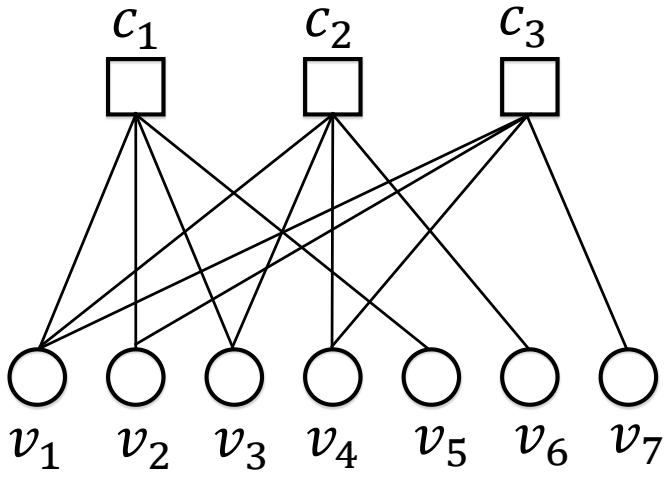


Figure 2.1: Tanner graph associated with binary parity check matrix in equation 2.1.

Figure 2.2 represents the Tanner graph of a length-5 LDPC code over $GF(4)$, with the parity-check matrix

$$H = \begin{bmatrix} \alpha & \alpha^2 & \alpha^3 & 0 & 0 \\ 0 & \alpha & \alpha^2 & \alpha^3 & 0 \\ 0 & 0 & \alpha & \alpha^2 & \alpha^3 \end{bmatrix}, \tag{2.2}$$

where α is the primitive element of the $GF(4)$. Since the parity-check matrix of a code is not unique, there can be different Tanner graphs representing the same code.

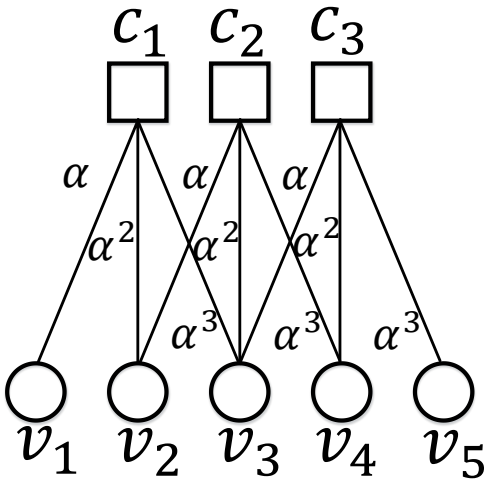


Figure 2.2: Tanner graph associated with non-binary parity check matrix in equation 2.2.

The rows of a parity-check matrix of a code define a set of constraints which each codeword of the code must satisfy. Thus, the parity-check matrix can be interpreted as a set of linear equations called the *parity-check equations*. Since any linear combination of the parity-check equations generates another valid parity-check equation, different parity-check matrices can represent the same code.

2.2 Decoding of Graph-Based Codes

Consider the block diagram of a communications system shown in Figure 2.3, where a linear block code is utilized to make the transmitting data resistant to error. Given the vector u of data, the encoder maps u to a valid codeword c, which is determined based on the parity-check matrix of the linear block code. At receiver, given the noisy realization from channel, r, the decoder attempts to form an estimate of the transmitted codeword. Various algorithm are introduced in academia for the decoding process of graph-based codes. We now briefly review the most well-known decoding approaches of graph-based codes, the maximum-likelihood decoding and the iterative message-passing decoding.

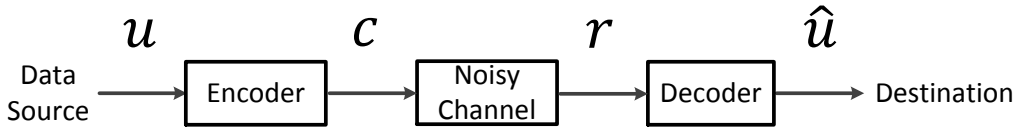


Figure 2.3: Block diagram of a generic communications system.

2.2.1 Maximum-Likelihood Decoding

In this section, we define notions of "optimal" decoding algorithms. Given a received noisy vector r from the channel, codeword $\hat{c}$ which maximizes $p(c|r)$ is most likely codeword to have been transmitted. If the channel is memoryless and each of the codewords is equally likely, then this reduces to the codeword $\hat{c}$ which maximizes $p(r|c)$. This is known as the maximum-likelihood (ML) estimate of the transmitted codeword.

Definition 2. *Given a code C and a received vector r, the maximum-likelihood decoder has as its output*

$$\hat{c} = \arg\max_{c \in C} p(c|r) \tag{2.3}$$

where $\hat{c}$ is the decoded codeword and arg max *is ranging over all codewords of the code C.*

An illustrative two-dimensional example for decoding process in ML decoders is shown in Figure 2.4, where for the received realization from the channel (black dot), the ML decoder maps this vector to the closest (most probable) codeword in the codeword space.

2.2.2 Iterative Message-Passing Decoding

Since maximum-likelihood decoder has an extremely high computational complexity, a lower complexity class of decoding algorithms are used in practice to decode graph-based codes. These sub-optimal decoders are collectively named message-passing al-

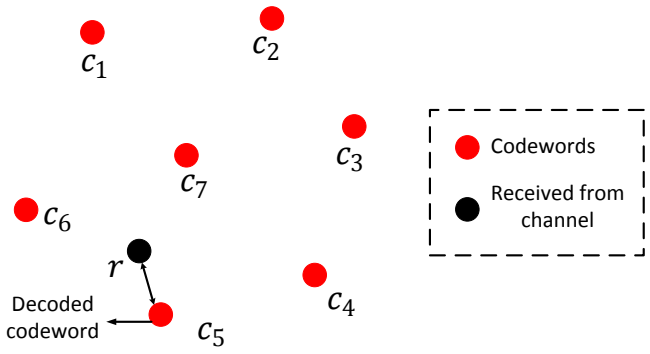


Figure 2.4: An illustrative example for maximum-likelihood decoding.

gorithms since their operation can be explained by the passing of messages along the edges of a Tanner graph. Each Tanner graph node works in isolation, only having access to the information contained in the messages on the edges connected to it. The message-passing algorithms are also known as iterative decoding algorithms as the messages pass back and forward between the variable and check nodes iteratively until a codeword is decoded or the maximum number of iterations is achieved. Several message-passing algorithms are introduced in academia which differ from each other due to either the type of messages passed or the type of operations performed at the variable/check nodes.

As an example of message-passing decoders, we briefly describe the well-known and simple bit-flipping decoder. In practical applications, more complicated but more accurate message-passing decoders, such as sum-product decoder and min-sum decoder, are usually used to decode graph-based codes.

The bit-flipping algorithm is a hard-decision message-passing algorithm for LDPC codes. A binary (hard) decision about each received bit is made by the detector and this is passed to the decoder. The main properties of the bit-flipping algorithm is as follows:

- **Messages:** For the bit-flipping algorithm the messages passed along the Tanner graph edges are binary: a variable node sends a message declaring if it is a one or a zero, and each check node sends a message to each connected bit node,

declaring what value the bit is based on the information available to the check node.

- **Check node computation:** Based on the values received from its neighboring variable nodes, each check node determines whether its parity-check equation is satisfied (i.e., if the modulo-2 sum of the incoming bit values is zero) or not.

- **Variable node computation:** If the majority of the messages received by a variable node are different from its received value the bit node changes (flips) its current value. This process is repeated until all of the parity-check equations are satisfied, or until some maximum number of decoder iterations has passed and the decoder gives up.

The bit-flipping decoder can be immediately terminated whenever a valid codeword has been found by checking if all of the parity-check equations are satisfied. This is true of all message-passing decoding of LDPC codes and has two important benefits; firstly additional iterations are avoided once a solution has been found, and secondly a failure to converge to a codeword is always detected. The bit-flipping algorithm is based on the principal that a codeword bit involved in a large number of incorrect check equations is likely to be incorrect itself. The sparseness of parity-check matrix in LDPC codes helps spread out the variable nodes into check nodes so that parity-check equations are unlikely to contain the same set of codeword bits.

2.3 Objects of Interest in Graph-Based Codes

It is well-known that for sufficiently high signal-to-noise ratios (SNRs), graph-based codes tend to exhibit sudden changes in the slope of the bit-error rate (BER) curve. Several researchers have investigated the problem of error floor in graph-based codes and introduced different graphical objects to characterize the origin of error floor in these codes. Trapping sets were first presented by Richardson in [23] to explain this

behavior. Several work provide analysis of trapping sets for binary and non-binary LDPC codes [17, 42]. More recently, Dolecek [16] introduced the notion of absorbing sets as the fixed points of message-passing decoders and provided analysis of absorbing sets for a class of regular LDPC codes called array-based codes. In the following subsections, we provide the mathematical definitions of binary tapping and absorbing sets.

2.3.1 Binary Trapping Set

Let $G = (V, C, F)$ denote a bipartite graph (Tanner graph) describing an LDPC code, with the usual notation of V being the set of bit nodes, C the set of check nodes, and the set F describing the edges between the nodes in V and C. For a given $D \subseteq V$, we say that a check node $c \in C$ is satisfied (unsatisfied) with respect to D if c is connected to D even (odd) number of times. For any subset $D \subseteq V$, let $E(D)$ (resp. $O(D)$) be the set of neighboring satisfied (unsatisfied) checks in C. The following provides the definition of trapping sets [17].

Definition 3. *For the graph $G = (V, C, F)$, an (a, b) trapping set $T_{a,b}$ is a subset D of V such that D contains a bit nodes and $O(D)$, $O(D) \subseteq C$, contains b check nodes.*

Moreover, an *elementary trapping set* [17] is a trapping set with each of its neighboring satisfied checks having exactly two edges connected to the trapping set, and each of its neighboring unsatisfied checks having exactly one edge connected to the trapping set.

2.3.2 Binary Absorbing Set

Since not all trapping sets are problematic in practical (finite-precision) iterative decoding algorithms, it is useful to characterize the subclass of trapping sets that is the main cause of errors under such decoders. Absorbing set [16] are combinatorial objects in Tanner graph that are guaranteed to be stable under a bit-flipping decoder.

Definition 4. *An $A_{a,b}$ binary absorbing set of size (a, b) is a subset $D \subseteq V$ with size a that connects to a subset $O(D) \subseteq C$ with size b, where each element of D has strictly fewer neighbors from $O(D)$ than from $C \backslash O(D)$.*

Clearly, there can exist even small trapping sets that do not fulfill the combinatorial requirements of absorbing sets, and therefore would not pose problems under bit-flipping decoder. The following statement rounds up the definitions of the graphical objects of interest.

Definition 5. *An elementary (fully) binary absorbing set is an (fully) absorbing set with each of its neighboring satisfied checks having two edges connected to the (fully) absorbing set, and each of its neighboring unsatisfied checks having exactly one edge connected to the (fully) absorbing set.*

2.4 Constructions of Structured LDPC Codes

In this section, we introduced the constructions of various binary and non-binary structured LDPC codes. In particular, we consider the constructions of binary array-based codes, non-binary protograph-based codes, and non-binary quasi-cyclic codes. Each of these constructions will be used in our discussions in the later chapters.

2.4.1 Binary Array-Based LDPC Codes

Array-based LDPC codes are a class of quasi-cyclic LDPC codes, which are first introduced in [43]. These codes are parameterized by their blocklength n and column weight c such that $c \leq p = \sqrt{n}$, where p is a prime. Given a $p \times p$ permutation matrix σ of the form

$$\sigma = \begin{bmatrix} 0 & 0 & ... & 0 & 1 \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 & 0 \end{bmatrix}, \tag{2.4}$$

we form the parity check matrix $H(c,p)$ of an $(n, n-c\sqrt{n}+c-1)$[1] array-based LDPC code as

$$H(c,p) = \begin{bmatrix} I & I & I & ... & I \\ I & \sigma & \sigma^2 & ... & \sigma^{(p-1)} \\ I & \sigma^2 & \sigma^4 & ... & \sigma^2(p-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & \sigma^{(c-1)} & \sigma^{2(c-1)} & ... & \sigma^{(p-1)(c-1)} \end{bmatrix}. \tag{2.5}$$

The matrix $H(c,p)$ can be viewed as a 2-D array of submatrices where each row (column) of matrices denotes a row (column) group $i, 0 \leq i \leq c-1$ ($j, 0 \leq j \leq p-1$). For the sake of our discussions, we describe each column of $H(c,p)$ by a pair (j,k) where j is the index of the column group, and k, $0 \leq k \leq p-1$, is the index of the column within the column group.

In Chapter 3, we provide the finite-length analysis of spatially-coupled codes which are constructed based on these array-based LDPC codes.

2.4.2 Non-Binary Protograph-Based LDPC Codes

There is a considerable freedom in choosing the edge weights in constructing protograph-based non-binary LDPC codes. In this dissertation, we consider the case where the edges are weighted independently of each other. We refer to resultant codes as *unconstrained* non-binary protograph-based (U-NBPB) codes.

[1]Note that the matrix $H(c,p)$ is not full rank.

A protograph $G = (V, C, E)$ [30] consists of the set $V = \{v_1, v_2, \ldots, v_{n_v}\}$ of variable nodes, the set $C = \{c_1, c_2, \ldots, c_{n_c}\}$ of check nodes, and the set $E = \{e_1, e_2, \ldots, e_{|E|}\}$ of edges connecting variable nodes and check nodes. Here, n_v is the total number of variable nodes, n_c is the total number of check nodes, and $|E|$ is the cardinality of the edge set E.

When the graph G is copied N times, each variable node $v_i \in V$ (each check node $c_i \in C$) in this mother protograph produces the set V_i of variable nodes $\{v_{i_1}, \ldots, v_{i_N}\}$ (the set C_i of check nodes $\{c_{i_1}, \ldots, c_{i_N}\}$) in the resultant daughter graph G^N. Likewise, each edge $e_i \in E$ in the mother protograph produces the set E_i of edges in the daughter graph where $E_i = \{e_{i,1}, \ldots, e_{i,N}\}$, and the edge $e_{i,j}$ for $1 \leq j \leq N$ connects the variable node v_{k_j} and the check node c_{l_j} if the edge e_i connects the variable node v_k and the check node c_l in the mother protograph. We denote the resultant daughter graph $G^N = (V^N, C^N, E^N)$.

We first provide the definition of U-NBPB codes and their ensembles. Let π_i denote the edge permutation associated with N copies of edge i.

Definition 6 (U-NBPB code)**.** *Given the mother protograph $G = (V, C, E)$, a $(G, N, \{s_k\}_k, \{\pi_i\}_i)$ U-NBPB code is constructed from the daughter graph $G^N = (V^N, C^N, E^N)$ by permuting the edges in the set E_i according to π_i for each $1 \leq i \leq |E|$, followed by scaling each edge k in G^N by a non-zero element s_k of $GF(q)$ for $1 \leq k \leq N \cdot |E|$.* ■

The U-NBPB code construction is illustrated in Figure 2.5(a) based on the mother protograph with $n_v = 3$, $n_c = 2$ and $N = 3$. The U-NBPB ensemble is defined as follows.

Definition 7 (U-NBPB code ensemble)**.** *The (G, N, q) U-NBPB ensemble is the collection of all $(G, N, \{s_k\}_k, \{\pi_i\}_i)$ U-NBPB codes with all possible choices of s_k's as non-zero elements of $GF(q)$ (for $1 \leq k \leq N \times |E|$) and $\{\pi_i\}$'s as all possible N-permutations (for $1 \leq i \leq |E|$).* ■

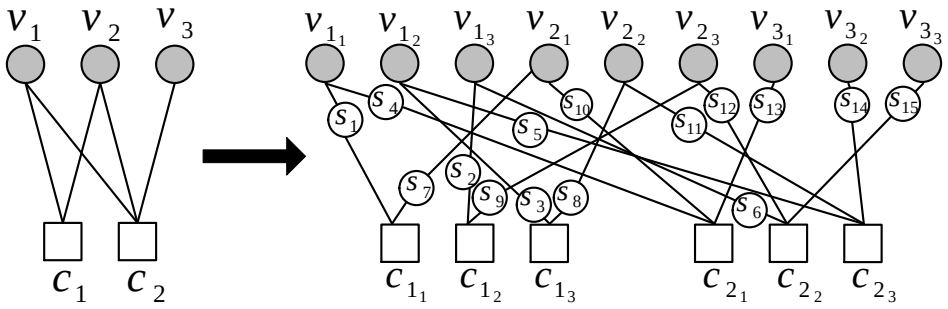


Figure 2.5: The original unlabeled protograph and an example of a U-NBPB code construction with $N = 3$ (copy-scale-permute).

In Chapter 4, we compute the codeword weight enumerators for U-NBPB codes, which are known to be useful for bounding the performance under the maximum-likelihood (ML) decoding.

2.4.3 Non-Binary Quasi-Cyclic LDPC Codes

Due to their implementation-friendly structure and superior performance, non-binary quasi-cyclic (NB-QC) LDPC codes are well-suited for emerging data storage applications requiring very low error rates. The following describes the construction of these codes. Assume that the following parameters are given:

- n_v: the number of columns in the binary base matrix,
- n_c: the number of rows in the binary base matrix,
- z: the lifting factor,
- c: the weight of each column in the binary base matrix,
- ℓ: the weight of each row in the binary base matrix.

The construction of a (c, ℓ)-regular NB-QC code over GF(q) using lifting involves the following steps [44]:

1) **Choosing the protograph**: The construction starts with the choice of an $n_c \times n_v$ binary parity-check matrix H with column weight c and row weight ℓ. The parity-check matrix H can be equivalently represented by a bipartite graph $G = (V, C, F)$ called the Tanner graph, with the usual notation of V being the set of variable nodes $v_i, i \in \{1, \ldots, n_v\}$, C being the set of check nodes $c_j, j \in \{1, \ldots, n_c\}$, and the set F describing the edges between the nodes in V and C. The Tanner graph G is called the protograph of the NB-QC code.

2) **Lifting the protograph**: The lifted matrix $\hat{H}$ is constructed by replacing each entry in matrix H with a $z \times z$ matrix. The zero entries in H are replaced by $z \times z$ zero matrices. Each non-zero entry corresponding to edge e in the protograph is replaced by a (z, d_e) circular permutation matrix (CPM). Here, (z, d_e) CPM refers to the $z \times z$ binary matrix obtained by circularly shifting the rows of the identity matrix by d_e places. Throughout the paper, d_e is called the lifting parameter associated with the edge e. The corresponding Tanner graph $\hat{G}$ of matrix $\hat{H}$ is called the binary lifted graph.

3) **Edge weight assignment**: In this step, non-binary weights are assigned to the edges of the binary lifted graph. Let α be a primitive element of GF(q). We choose a parameter λ such that $(q-1)|\lambda z$. We also select a parameter $\rho_e \in \{0, 1, \ldots, q-2\}$ for each edge e in the protograph G. Then, the value of the non-zero element in the kth row of the (z, d_e) CPM is replaced by $\alpha^{\rho_e + (k-1)\times\lambda}$. Throughout the paper, we refer to ρ_e as the labeling parameter associated with the edge e.

Note that the resulting NB-QC code has the length of $n_v z \log_2 q$ bits and the design rate of $\frac{n_v - n_c}{n_v}$. The construction introduced in [45] is a special case of the NB-QC construction using lifting when $z = q-1$, $\lambda = 1$ and $d_e = \rho_e$ for any edge e. The code designs in [46] and [47] are also special cases of the NB-QC construction using lifting.

CHAPTER 3

Finite-Length Analysis and Design of Non-Binary Block LDPC Codes

It is empirically shown that non-binary LDPC codes perform significantly better than their binary counterparts in moderate and high SNR regions. Therefore, non-binary LDPC systems are considered as great candidates for emerging data storage systems. But there are two major issues facing utilization of non-binary LDPC systems. First, these codes, similar to binary LDPC codes, suffer from the error floor. Second, the message passing algorithms which are currently used to decode non-binary LDPC codes have extremely high computational complexity. In this section, we try to address the first problem.

This chapter characterizes the error floor problem for non-binary LDPC codes and consequently provide guidelines for designing non-binary codes with superior error floor performance. In particular, our optimized unstructured and structured codes show at least one order of magnitude error floor performance improvement compared to unoptimized codes. Utilizating of these codes in practice can result in considerable improvement in capacity and lifetime of data storage systems.

3.1 Introduction

For the case of non-binary LDPC codes, not much is known about how non-codeword objects and specific substructures in the Tanner graph affect the error floor performance. In [18], the authors show that small cycles in the Tanner graph of column weight two

codes (i.e., the cycles that correspond to codewords with a low minimum distance) significantly degrade the code performance in the error floor region. An effective approach to improve the error floor performance of column weight two codes based on cycle manipulations is also introduced in [18]. Further, in the case of the binary erasure channel (BEC), recent results include the introduction of the peeling decoder and the (generalized) stopping sets for non-binary LDPC codes [19]. In [24], the stopping constellation distributions for irregular non-binary LDPC code ensembles was computed. Further, zig-zag cycles and their relationship with the error floor were analyzed in [18, 25, 26]. Ensemble enumerators for both stopping and trapping sets were computed in [12, 27] for protograph-based ensembles of non-binary LDPC codes. However, apart from the recent result in [28], where a simplified absorbing set definition (compared to the one in [16]) is given, no other results are known for non-binary absorbing sets.

The main goals of this chapter are multifold: 1) to define and analyze non-binary absorbing sets for codes over GF(q), $q > 2$, 2) to highlight the difference between binary absorbing sets and non-binary absorbing sets, and 3) to propose an efficient code design based on our classification of absorbing sets.

3.2 Non-Binary Absorbing Sets

In contrast to binary codes, each edge in the Tanner graph of a non-binary code admits a weight taken as a non-zero element of the underlying non-binary field. Consequently, the edges of the subgraphs which correspond to the fixed points of non-binary LDPC decoders need not only be topologically connected in specific ways, but also the labels on these edges must satisfy certain conditions. In other words, suppose we consider a topology (with no edge weight assignment) that satisfies the conditions of the (binary) absorbing set, as given by Definition 4. Once the edge labels, taken over some field GF(q), are chosen, the resulting object, when interpreted over GF(q), may or may not cause a decoding error.

Example 1 illustrates the difference between binary and non-binary absorbing sets and gives a motivation for the definition of non-binary absorbing sets.

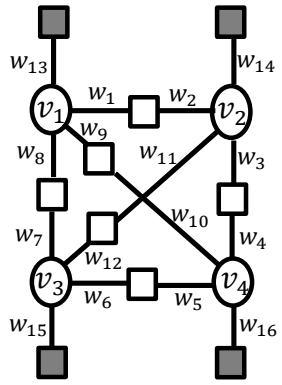


Figure 3.1: Tanner graph of a $(4, 4)$ absorbing set.

Example 1. *Consider the graphical structure in Figure 3.1, defined over GF(q). If there exists a set of non-zero inputs for all variable nodes that makes all degree-two check nodes satisfied, the resulting configuration will have* 4 *unsatisfied checks and each variable node will have strictly more satisfied than unsatisfied neighboring checks (3 vs. 1). Mathematically, the inputs* v_1, v_2, v_3, v_4 *and weights of the edges* $w_1, \ldots, w_{12}$ *then satisfy:*

$$v_1 w_1 = v_2 w_2 \quad \text{over GF}(q), \quad v_2 w_3 = v_4 w_4 \quad \text{over GF}(q), \quad v_4 w_5 = v_3 w_6 \quad \text{over GF}(q),$$
$$v_3 w_7 = v_1 w_8 \quad \text{over GF}(q), \quad v_2 w_{11} = v_3 w_{12} \quad \text{over GF}(q), \quad v_1 w_9 = v_4 w_{10} \quad \text{over GF}(q),$$

which leads to the following conditions:

$$w_1 w_7 w_{11} = w_2 w_8 w_{12} \quad \text{over GF}(q), \quad w_3 w_5 w_{12} = w_4 w_6 w_{11} \quad \text{over GF}(q),$$
$$w_2 w_4 w_9 = w_1 w_3 w_{10} \quad \text{over GF}(q), \tag{3.1}$$

where all w *and* v *are non-zero elements of GF(q).*

For example, for $q = 8$*, Figure 3.2(a) shows a choice of weights satisfying the conditions in* (3.1)*. With these weights, there are* $q - 1$ *choices (out of* $(q-1)^4$*) for the*

set (v_1, v_2, v_3, v_4) *such that each variable node has exactly 3 satisfied and 1 unsatisfied checks. One example is* $(1, 2, 4, 1)$. *In contrast, the weights in Figure 3.2(b) do not satisfy the conditions in* (3.1) *and result in a configuration that has* 4 *variable nodes and* z *unsatisfied checks. Here,* $5 \leq z \leq 10$*, and the value of* z *depends on the input values* v_1 *through* v_4*. For example, the same input* $(1, 2, 4, 1)$ *results in* $z = 10$*. Clearly, this configuration with all* 10 *neighboring checks being unsatisfied is not expected to be problematic for belief propagation decoding.*

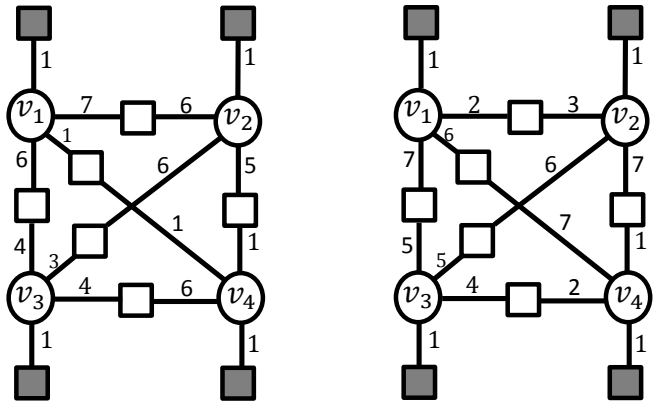


Figure 3.2: (a) Non-binary elementary $(4, 4)$ absorbing set over GF(8), (b) $(4, z)$ trapping set, $5 \leq z \leq 10$, weights do not satisfy absorbing set conditions over GF(8).

Example 1 above provides a motivation for studying non-binary absorbing sets. Consider a non-binary LDPC code with an $m \times n$ parity check matrix H defined over GF(q). The corresponding Tanner graph has n variable nodes and m check nodes. Definition 8 below states the conditions for a subset of variable nodes to be an (a, b) non-binary absorbing set. We assume transmission of the all-zero codeword. We also assume that these a decoding errors only occur in variable nodes included in this absorbing set, and that values of the variable nodes outside of the absorbing set are all 0 ($\in$ GF(q)).

Consider a subset $\mathcal{V}$ of variable nodes with $|\mathcal{V}| = a$. We form the $\ell \times a$ matrix

A, a submatrix of matrix H, consisting of the columns of matrix H that correspond to variable nodes in $\mathcal{V}$ and ℓ check nodes connected to $\mathcal{V}$.

Definition 8. *The configuration $\mathcal{V}$ is an (a, b) absorbing set over GF(q) if there exists an $(\ell - b) \times a$ submatrix B of rank r_B, with elements $b_{j,i}, 1 \leq j \leq \ell - b, 1 \leq i \leq a$, in matrix A that satisfies the following conditions.*

1. *Let $N(B)$ be the null-space of matrix B and let $\boldsymbol{d}_i, 1 \leq i \leq b$ be the ith row of matrix D, where D is formed by excluding the matrix B from A. Then,*

$$\exists \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_a \end{bmatrix} \in N(B) \text{ s.t. } x_i \neq 0 \text{ for } \forall i \in \{1, \ldots, a\} \text{ and } \nexists i, \boldsymbol{d}_i \boldsymbol{x} = 0. \tag{3.2}$$

2. *Let $d_{j,i}$, $1 \leq j \leq b, 1 \leq i \leq a$, be the elements of the matrix D. Then,*

$$\forall i \in \{1, 2, \ldots, a\} : \left(\sum_{j=1}^{\ell - b} S(b_{j,i}) \right) > \left(\sum_{j=1}^{b} S(d_{j,i}) \right), \tag{3.3}$$

where the function S is

$$S(x) = \begin{cases} 1 \text{ when } x > 0, \\ 0 \text{ when } x \leq 0. \end{cases} \tag{3.4}$$

Condition 1 in Definition 8 requires that there exists a vector in the null-space of the matrix B all of whose elements are non-zero. Therefore, there exists a solution to $Bx = 0$ over GF(q) such that all components of the solution are non-zero. A consequence of this condition is that $r_B < a$. Also, condition 1 guarantees that for vector $\boldsymbol{x}$ in the null-space of matrix B, none of the check nodes associated with the rows of the matrix D are satisfied (otherwise, input $\boldsymbol{x}$ results in an $(a, \tilde{b})$, $\tilde{b} < b$, absorbing set). Condition 2 ensures that for each variable node, the number of connected satisfied checks is larger than the number of connected unsatisfied checks.

Observe that the proposed definition is in agreement with the existing definition of a binary absorbing set [16]. In particular, for $q = 2$, the condition 1 is automatically satisfied since the variable nodes' input vector must be an all-ones vector to satisfy the checks. Also, the condition that each variable node in the absorbing set has strictly more neighbors in $\mathcal{E}$ than in $\mathcal{O}$ corresponds to condition 2.

Remark 1. *Our definition of a non-binary absorbing set is different from the definition proposed in [28] where the authors define a non-binary (primitive) absorbing set as an object which has more satisfied checks than unsatisfied checks, taken collectively over all variable nodes in this object. In contrast, our definition, similar to the original definition of binary absorbing sets [16], requires* each *variable node to be connected to more satisfied checks than unsatisfied checks.*

Remark 2. *Note that whether a non-binary absorbing set as defined in Definition 8 results in a decoding error depends on the choice of the variable nodes inputs. The set of all* $\boldsymbol{x}$*'s which satisfy* (3.2) *is the set of all variable nodes inputs that result in an* (a, b) *non-binary absorbing set. Other choices of variable nodes inputs results in other* $(a, \tilde{b})$*,* $\tilde{b} \neq b$ *absorbing/trapping sets.*

As in the binary case, we say that a non-binary absorbing set $\mathcal{V}$ is **elementary** if all neighboring satisfied checks have degree 2 and all neighboring unsatisfied checks have degree 1 with respect to $\mathcal{V}$. It can be easily observed that an elementary non-binary absorbing set is necessarily an elementary binary absorbing set where all non-zero edge labels and non-zero variable node values are converted to 1 and all operations are taken modulo 2. This observation can be used in searching for elementary non-binary absorbing sets in the Tanner graph of non-binary codes. Clearly, the converse is not true as the choice of non-binary labels may violate the absorbing set constraints.

In the case of elementary absorbing sets, the absorbing set conditions can be simplified, as the following lemma shows. Consider a code $\mathcal{C}$ with a parity check matrix H over GF(q). Let $\mathcal{G}$ be its Tanner graph. Let C_p be an arbitrary cycle involving p

distinct variable nodes and p distinct neighboring check nodes in the graph induced by an elementary (a, b) non-binary absorbing set in $\mathcal{G}$, $p \leq a$. We write C_p as the oriented traversal $c_1 - v_1 - c_2 - v_2 - \cdots - c_p - v_p - c_1$, where v and c denote the spanned variable and check nodes, respectively. Let w_{2i+1} denote the label on the $c_i - v_i$ edge, and let w_{2i} denote the label on the $v_i - c_{i+1}$ edge. The following lemma presents a necessary condition for a subgraph of the Tanner graph of the code $\mathcal{C}$ to be an elementary non-binary absorbing set.

Lemma 1. *In the case of elementary absorbing sets over GF(q), for every cycle C_p (as introduced above), the weights of the edges w_i, $i \in \{1, 2, \ldots, 2p\}$, satisfy the following relation:*

$$\prod_{k=1}^{p} w_{2k-1} = \prod_{k=1}^{p} w_{2k} \quad \text{over GF}(q). \tag{3.5}$$

Proof. For the cycle C_p (of length $2p$) we form a $p \times p$ submatrix B_{C_p} of H, corresponding to the p variable nodes and p check nodes in C_p. Since the check nodes in B_{C_p} are satisfied, there exists a non-zero solution x to $B_{C_p} x = 0$ over GF(q). Therefore, for the square matrix B_{C_p}, we have det$(B_{C_p}) = 0$ over GF(q). Thus,

$$\det(B_{C_p}) = \det \begin{bmatrix} w_1 & w_2 & 0 & ... & 0 \\ 0 & w_3 & w_4 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & w_{2p-3} & w_{2p-2} \\ w_{2p} & 0 & ... & 0 & w_{2p-1} \end{bmatrix} = 0 \quad \text{over GF}(q). \tag{3.6}$$

Now, in order to satisfy (3.6), the condition in (3.5) must hold. Since every permutation of the columns of B_{C_p} results in same determinant, the chosen ordering was performed without loss of generality. □

It can be shown that if all the fundamental cycles satisfies (3.5), this also holds for all other cycles in the graph. Thus, an elementary non-binary absorbing set not only satisfies the topological conditions, i.e., the unlabeled subgraph is an elementary *binary* absorbing sets, but also all of its fundamental cycles satisfy (3.5).

Remark 3. *A condition similar to Lemma 1 is presented in [18] but only for regular codes of column weight* 2*. In the case of column weight* 2*, the smallest absorbing sets are* $(a, 0)$ *absorbing sets which correspond to the minimum weight codewords, and moreover contain only one (fundamental) cycle. The results presented in this paper apply to general column weights wherein absorbing sets may be spanned by more than one fundamental cycle.*

Example 2. *Let us interpret the configuration in Figure 3.1 as a* $(4, 4)$ *non-binary absorbing set. The conditions on the cycles in the cycle span (containing 3 cycles) in* (3.5) *are precisely the one given in* (3.1)*. If, however,* (3.5) *is violated for one of the three fundamental cycles, the resulting configuration will no longer be an absorbing set; it will instead become a* $(4, z), 5 \leq z \leq 12$*, trapping set (where the value of* z *depends on the input and on the number of unsatisfied checks).*

Lemma 2. *Based on the non-binary edge weights and the variable node inputs, an unlabeled* (a, b) *(i.e., binary) elementary absorbing set becomes an* (a, b^+) *absorbing set/trapping set in the non-binary case (*b^+ *denotes any integer greater than or equal to* b*).*

Proof. Based on Definition 5, elementary unlabeled (binary) absorbing sets include only degree two and degree one check nodes. After the assignment of non-binary edge weights, degree one check nodes always remain unsatisfied since the multiplication of two non-zero Galois field elements is always non-zero. Based on the inputs of the variable nodes and the edge weights, degree two check nodes may remain satisfied or may become unsatisfied. Therefore, after the assignment of non-binary edge weights, the number of unsatisfied checks either stays the same or increases. □

Remark 4. *In Definition 8, transmission of the all-zeros codeword is assumed without loss of generality. In transmission of any other codeword, because of the linearity of the code, addition of the same error vector* $\boldsymbol{x}$ *(defined in Definition 8, Condition* 1*) results in the same absorbing set error.*

3.3 Non-Binary Absorbing Sets as a Tool to Improve the Performance

In this section, equipped with techniques from graph theory, we find the ratio of all possible edge weight assignments which convert an (a, b) unlabeled elementary absorbing set to an (a, b) non-binary elementary absorbing sets or to an $(a, b + 1)$ non-binary trapping set in GF(q). Furthermore, based on the observation that proper choices of non-binary edge weight assignments result in non-problematic trapping sets, we propose algorithms to improve the performance of non-binary unstructured and non-binary quasi-cyclic LDPC codes in the error floor region.

Theorem 1. *Given an (a, b) unlabeled (i.e., binary) elementary absorbing set with e satisfied checks:*

1. *A fraction of $(q-1)^{a-e-1}$ out of all possible edge weight assignments taken from GF(q) results in (a, b) elementary non-binary absorbing sets.*
2. *A fraction of $e \cdot (q-1)^{a-e-1} \cdot (q-2)$ out of all possible edge label assignments taken from GF(q) results in $(a, b+1)$ non-binary trapping sets.*

Proof. The VN graph of the given unlabeled elementary absorbing set includes a vertices and e edges. Each edge in the spanning tree of the VN graph corresponds to two edges in the Tanner graph.

For part 1, the labels of the edges in the Tanner graph that are represented in the spanning tree can be chosen arbitrarily. Therefore, we have $(q-1)^2$ choices for the weights for each edge. Thus, there are $(q-1)^{2(a-1)}$ weight assignments for the $a-1$ edges in the spanning tree.

For each of the remaining $e-(a-1)$ edges in the VN graph that are not in the spanning tree, one of the two edges in the Tanner graph can again be chosen arbitrarily but the other edge is uniquely determined according to (3.5). Thus, there are $(q-$

$1)^{e-a+1}$ weight assignments for edges not in the spanning tree. Hence, the total number of the weight assignments resulting in elementary non-binary absorbing sets is given as

$$(q-1)^{2(a-1)} \cdot (q-1)^{e-a+1} = (q-1)^{a+e-1}.$$

Since, we have a total of $(q-1)^{2e}$ possible weight assignments, the resulting fraction follows.

For part 2, in addition to b degree one unsatisfied checks, one of the degree two check nodes is also unsatisfied. Therefore, equation (3.5) should be satisfied for all fundamental cycles except the one which includes the additional unsatisfied check node. We consider a choice of a spanning tree which does not include the edge representing the degree two unsatisfied check node in the VN graph. Similar to part 1, the edge weights in the Tanner graph that are included in the spanning tree can be chosen arbitrarily. Thus, there are $(q-1)^{2(a-1)}$ weight assignments for the $a-1$ edges in the spanning tree.

For each of $e-(a-2)$ edges in the VN graph which represent satisfied checks outside of the spanning tree, one of the two edges in the Tanner graph can again be chosen arbitrarily but the other edge is uniquely determined according to (3.5). Thus, there are $(q-1)^{e-a}$ weight assignments for edges which represent satisfied checks not in the spanning tree of the VN graph. For the remaining edge in the spanning tree (which corresponds to the degree two unsatisfied check node), one of the two edges in the Tanner graph can be chosen arbitrarily ($q-1$ choices). The other edge in the Tanner graph should be chosen from $q-2$ weights, which results in a violation of (3.5) for the corresponding fundamental cycle. Hence, the total number of the weight assignments resulting in $(a, b+1)$ non-binary trapping sets is given as

$$\binom{e}{1}(q-1)^{2(a-1)} \cdot (q-1)^{e-a} \cdot (q-1) \cdot (q-2) = e(q-1)^{a+e-1}(q-2).$$

Note that the multiplication by $\binom{e}{1}$ is due to the choice of one degree-two check node (from a total of e degree two check nodes) to be unsatisfied. Since we have a total of $(q-1)^{2e}$ possible weight assignments, the resulting fraction follows. □

From Theorem 1, we can conclude that compared to the number of (a, b) non-binary absorbing sets, a by a factor of $\binom{e}{1}(q-2)$ larger number of $(a, b+1)$ non-binary trapping sets exists in the Tanner graph of a randomly generated code. However, our simulation results will show that the error profiles of these codes do not include any errors of this type. Therefore, our proposed absorbing set definition provides a better description of the problematic subgraphs for the BP decoders in the error floor region.

3.3.1 Code Design Guidelines for Non-Binary Unstructured LDPC Codes

In this section, we will introduce a method to eliminate problematic non-binary absorbing sets from Tanner graphs of non-binary unstructured LDPC codes. Our simulation results will show the effectiveness of our algorithm in improving the code performance in error floor region.

We exploit the fact that non-binary edge weights enable us to reduce the number of absorbing sets by just changing the weights of edges in the Tanner graph without changing its structure. The method is stated in Algorithm 1. We say that an absorbing set A_y is defined as a child of an absorbing set A_z if A_z is a subgraph of A_y. The main steps of the method can be summarized as follows:

- Step 1: For the given Tanner graph, we first choose a set W of pairwise parameters (i.e., $W = \{(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\}$) corresponding to the k elementary absorbing sets which we want to eliminate.

- Step 2: We find the absorbing set (a_j, b_j) with the smallest a and b parameters in W. If the (a_j, b_j) absorbing set is a child of a previously eliminated absorbing set, we remove (a_j, b_j) from W and repeat this step for the next smallest parameters.

- Step 3: We find the set U_j of all binary (a_j, b_j) absorbing sets in the unlabeled Tanner graph. For each absorbing set in U_j, if all fundamental cycles satisfy (3.5), we change the weight of an edge to another non-zero element of GF(q).

The edge and its new weight are chosen such that the previously canceled non-binary absorbing sets remain canceled. This process continues until all (a_j, b_j) absorbing sets are eliminated. Then, we remove (a_j, b_j) from W.

- Step 4: If $W \neq \varnothing$, we go to Step 2. Otherwise, the algorithm terminates.

Remark 5. *The work in [18] focuses on the case of column weight two and presents an approach to cancel all cycles of length l, $g \leq l \leq l_{max}$, where g is the girth, for this column weight choice. As stated in [18], it is impossible to cancel all cycles for all lengths l. In contrast, for codes with column weights $c \geq 2$, our approach only seeks to cancel a selected number of cycles, i.e., one fundamental cycle per absorbing set of interest. As a result, our approach allows for further flexibility in canceling cycles of various lengths for column weights $c \geq 2$.*

Figure 3.3 shows the simulation results[1] for random regular codes (denoted by 'Original') over various choices of the field order size GF(q), with block length $N \approx 2750$ bits, rate $R \approx 0.88$, column weight $c = 4$, row weight $r = 51$ for GF(2), $r = 37$ for GF(4), $r = 31$ for GF(8), and $r = 26$ for GF(16), and girth $g = 6$, transmitted over a binary-input additive white Gaussian noise (AWGN) channel, where the frame error rate (FER) versus the signal-to-noise ratio (SNR) is displayed. The figure also shows the results obtained by using a modified version of the approach presented in [18] (denoted by 'P-method') which tries to cancel all the cycles of length l, $g \leq l \leq l_{max}$. The approach in [18], which was introduced for the case of $c = 2$, can easily and successfully be extended to the case of $c > 2$, as demonstrated in this paper. The figure also shows the results obtained by the code modification proposed in [48] (denoted by 'N-method') and our code modification specified in Algorithm 1 (denoted by 'A-method'). The simulations results reconfirm the superior performance of the codes modified by

[1] All the simulation results presented in this paper were performed over a six month period on the Hoffman2 Cluster which is a part of High Performance Computing Resources at UCLA. The Hoffman2 cluster has more than 800 machines and about 7000 cores. The CPUs have 8, 12 or 16 cores with speed of $2.2 - 3.0$ GHz. Each core has 1GB, 4GB or 8GB of memory. The Hoffman2 cluster uses the CentOS Linux 6.2 Operating System.

Algorithm 1 Reduction of the number of absorbing sets in the Tanner graph of a non-binary code

1: **Input:** Tanner graph T with edge weights over GF(q).
2: Choose W, the set of non-binary absorbing set to be eliminated.
3: Let X be the set of non-binary absorbing set which can not be eliminated.
4: Let $X = \varnothing$.
5: Let set $A = \varnothing$.
6: For every edge $j \in T$, C_j is the set of canceled cycles which include j.
7: $\forall j \in T$, $C_j = \varnothing$.
8: Find (a_j, b_j), the smallest non-binary absorbing set in $W \backslash A$.
9: If this absorbing set is a child of another absorbing set in A, go to 22.
10: Find U_j, the set of all (a_j, b_j) absorbing sets in unlabeled Tanner graph T.
11: **for** $\forall u \in U_j$ **do**
12: Find F_u, a set of fundamental cycles of u.
13: Let E_u be the set of all edges in u.
14: For an edge $k \in E_u$, let M_k be the set of cycles in F_u which include k.
15: **if** (3.5) is satisfied for all cycles in F_u **then**
16: If $E_u = \varnothing$, $X \leftarrow X \cup U_j$ and go to 22.
17: Find edge $i \in E_u$ with minimum $|C_i|$.
18: $w_i \leftarrow w_i + v$ with $v \neq w_i$ and $v \neq 0$ such that all cycles in C_i do not satisfy (3.5). If no v exists, $E_u \leftarrow E_u \backslash i$, and go to 17.
19: For every edge e in the cycles of M_i, update $C_e \leftarrow C_e \cup M_e$.
20: **end if**
21: **end for**
22: Add (a, b) absorbing set to the set A.
23: If $A \neq W$, go to 8.
24: If $X = \varnothing$, all absorbing sets of interest are eliminated. Otherwise, it is not possible to eliminate absorbing sets in X.

Table 3.1: Error Profile, $SNR = 5.2$ dB, $N = 2738$, $R = 0.891$, $c = 4$ and GF(4), total number of simulations $\approx 4 \times 10^8$.

Error Type	$(4,4)$	$(5,0)$	$(5,2)$	$(6,2)$	$(6,4)$	$(6,6)$	$(7,4)$	$(8,2)$	other
Original	32	9	14	7	9	18	9	9	15
P-method	0	0	0	0	0	12	5	5	12
A-method	0	0	0	0	0	0	0	0	13

N-method, compared to the codes modified by P-method, as shown previously in [48]. All codes are decoded by using a Fast Fourier Transform-based q-ary SPA (FFT-QSPA) decoder [14].

All P-method, N-method and A-method approaches enjoy the improved performance relative to the random code construction. The performance comparison for different values of q reveals that the improvement is more pronounced for smaller values of q. As we will show below in Section 4, Corollary 1, for a random code construction, under higher field sizes, there are fewer non-binary elementary absorbing sets available to be canceled by using the P-method, N-method, or the A-method. Table 3.1 includes the error profiles[2] for original, P-method and A-method codes over GF(4). Both the P-method and the A-method eliminate all $(4,4)$ absorbing sets as well as $(5,0)$, $(5,2)$, $(6,2)$ and $(6,4)$ absorbing sets that are children of $(4,4)$ absorbing sets. Additionally, the A-method successfully eliminates all $(6,6)$, $(7,4)$ and $(8,2)$ absorbing sets, since in this approach we selectively reweigh some (but not necessarily all) length-6 cycles followed by a reweighting of some of length-8 cycles. As q increases, there are more degrees of freedom available to change the edge weights to reweigh cycles and as a result, the implementation of the cycle-elimination-only approach in [18] appears to be sufficient for larger values of q. The errors labeled as 'other' in Table 3.1 are non-

[2]The error profiles are calculated as follows. After the decoder reaches its maximum number of iterations, for each channel realization which results in a decoding error, we form the induced subgraph corresponding to the variable nodes in error. We then determine whether the induced subgraph is an absorbing set or not. If yes, the size of the absorbing set is also calculated. In a small minority of cases, the induced subgraph includes two or more separate subgraphs which are independently investigated.

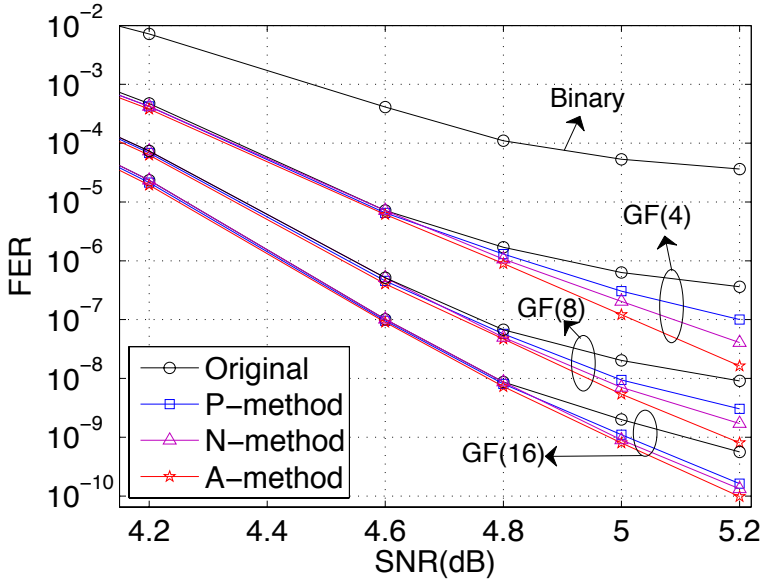


Figure 3.3: FER versus SNR for the original random non-binary code and for both P- and A-method modified codes, $N \approx 2750$, $R \approx 0.88$, $c = 4$, and the QSPA-FFT decoder.

converging errors. This type of error happens when the decoder does not converge to a specific object before reaching its maximum number of iterations. In other words, we declare "convergence" if the decision of the decoder does not change over 5 final iterations of decoding (out of 50 total iterations). If the decoder decision changes, the error is categorized as a non-converging error. This type of error is usually an oscillation between different trapping set errors.

Note that by performing our algorithm, (a, b) absorbing sets in the error profile of the original code are converted to structures which can turn into (a, b^+) trapping sets under appropriate variable node input values. As the error profile in Table 3.1 shows, these new structures are not problematic for the decoder. Therefore, non-binary absorbing sets provide a better definition for problematic structures in the error floor region compared to trapping sets.

The simulation results for random regular codes ('Original') and the modified codes using the A-method over GF(q), $q = 2, 4, 8, 16$ with block length $N \approx 2350$ bits, rate

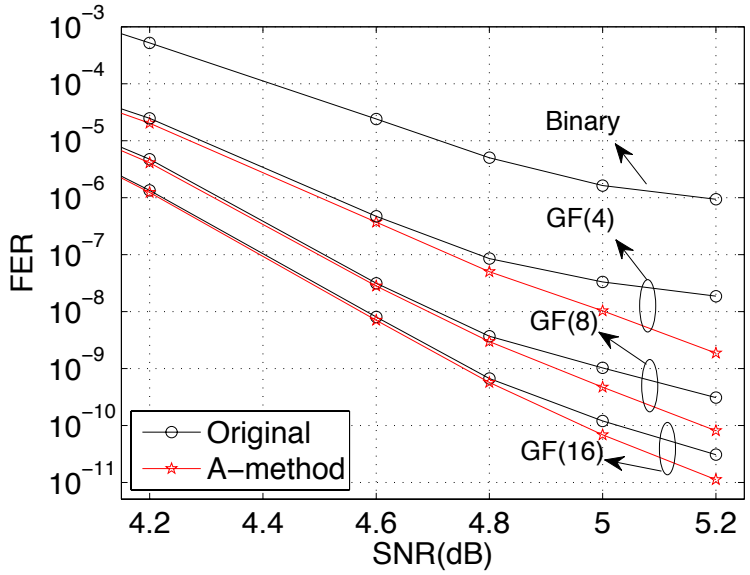


Figure 3.4: FER versus SNR for the original random non-binary code and the A-method modified code, $N \approx 2350$, $R \approx 0.83$, $c = 5$, and the QSPA-FFT decoder.

$R \approx 0.83$, column weight $c = 5$, and row weight $r = 50$ for GF(2), $r = 33$ for GF(4), $r = 28$ for GF(8), $r = 24$ for GF(16) are presented in Figure 3.4. This figure again confirms the effectiveness of our algorithm in improving the performance of non-binary codes in the error floor region. Similar improvements in performance were also observed by using the P-method, although those results have been omitted from Figure 3.4 for brevity. These simulations again show that the gap between the curves for the P-method and A-method decreases for larger field sizes, and that the improvement in the performance of the codes decreases as the Galois field size increases.

Figure 3.5 presents the simulation results for random regular codes ('Original') and modified codes using the P-method and the A-method over GF(q), $q = 2, 4, 8, 16$ with block length $N \approx 2750$ bits, rate $R \approx 0.88$, column weight $c = 4$, row weight $r = 51$ for GF(2), $r = 37$ for GF(4), $r = 31$ for GF(8), $r = 26$ for GF(16) and girth $g = 6$ using a non-binary min-sum decoder [15]. The improvement in the performance of the code achieved by both P- and A-methods can also be observed for the min-sum decoder. Furthermore, Table 3.2 shows the error profile of these three codes over GF(4)

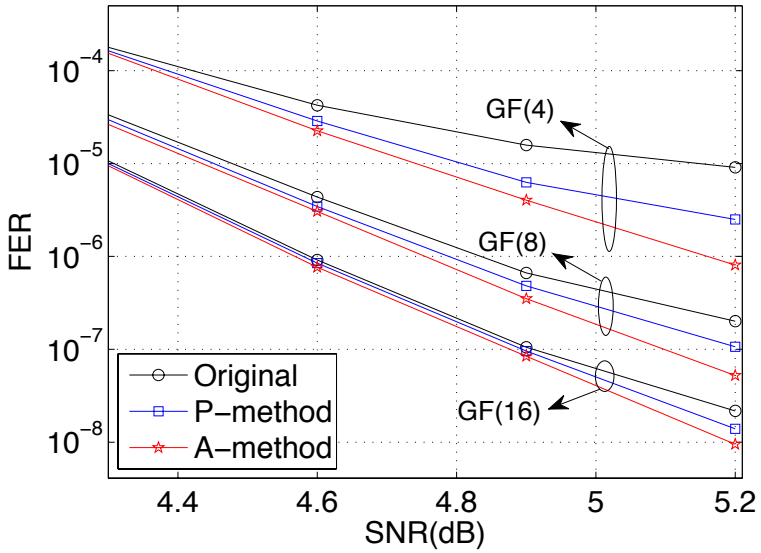


Figure 3.5: FER versus SNR for the original non-binary random code and both P- and A-method modified codes, $N \approx 2750$, $R \approx 0.88$, $c = 4$, and a min-sum decoder.

Table 3.2: Error Profile, $E_b/N_0 = 5.2\,\text{dB}$, $N = 2738$, $R = 0.891$, $c = 4$ and GF(4), total number of simulations $\approx 1.5 \times 10^7$.

Error Type	$(4,4)$	$(5,0)$	$(5,2)$	$(6,2)$	$(6,4)$	$(6,6)$	$(7,4)$	$(8,2)$	other
Original	43	9	14	8	8	12	7	6	23
P-method	0	0	0	0	0	9	5	4	14
A-method	0	0	0	0	0	0	0	0	15

at SNR=5.2 dB using the min-sum decoder. Although both P- and A-methods decrease the number of problematic absorbing sets in the original code, our proposed algorithm (A-method) is more effective as it cancels all small absorbing sets in the error profile.

Figure 3.6 shows the performance of non-binary quasi-cyclic codes (QC) [49] ('Original') with block length $N \approx 1200$, rate $R \approx 0.8$ and column weight $c = 4$ as well as modified codes using our proposed algorithm (A-method). This example shows that this algorithm works well independently of the code structure as it is effective for both structured and random codes. We again mention that the P-method also provides an improvement in the performance in error floor region. Similar to our previous examples,

the gap between the curves for A- and P-methods diminishes for larger field sizes.

It is established that irregular non-binary codes over small field sizes provide better error correcting performance compared to regular non-binary codes [50], [51], and [52]. Since our proposed algorithm is not limited to any specific construction of LDPC codes, it can also be utilized for irregular codes. Figure 3.7 shows the performance of non-binary irregular codes constructed using the Progressive Edge-Growth (PEG) approach [51] ('Original') with block length $N \approx 2000$ bits, rate $R \approx 0.85$ and variable node degree distribution[3] $\Lambda(x) = 0.5x^4 + 0.5x^5$ as well as modified codes using our proposed algorithm (A-method). This example reconfirms the effectiveness of our proposed algorithm in improving the performance of various code constructions in the error floor region.

Our various examples show that the proposed algorithm consistently improves the performances of the original codes across different choices of codes (different block lengths, column weights, and structures) and different decoder implementations (FFT-QSPA and min-sum decoders). These examples also show that our proposed definition of non-binary absorbing set appears to be applicable to any non-binary code regardless of the choice of the decoder.

3.3.2 Code Design Guidelines for Non-Binary Quasi-Cyclic LDPC Codes

In this section, we first analyze the necessary conditions for the existence of a non-binary elementary AS in the Tanner graph of a NB-QC code. We investigate how the topological and weight conditions map to certain equations which include the design parameters of NB-QC codes. This analysis enables us to propose an algorithm to design NB-QC codes with good error floor performance. In our proposed algorithm, we design NB-QC codes with reduced number of problematic elementary ASs by violating either

[3]Note that the chosen variable node degree distribution is obtained using the PEG approach in [51], but it is not optimal in terms of girth. The optimality of the variable node degree distribution in this example is not necessary for demostrating the effectiveness of our proposed algorithm; the example simply shows that the proposed algorithm works for irregular codes.

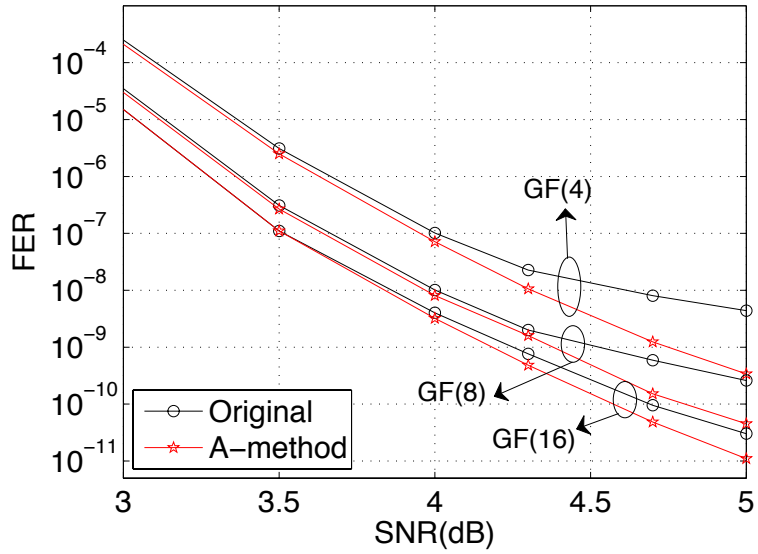


Figure 3.6: FER versus SNR for non-binary QC codes and their A-method modified versions, $N \approx 1200$, $R \approx 0.8$, $c = 4$, and the QSPA-FFT decoder.

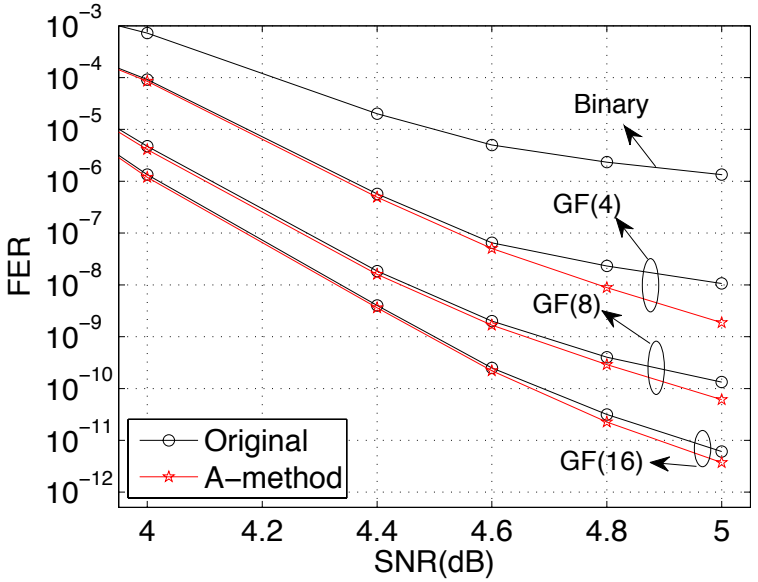


Figure 3.7: FER versus SNR for non-binary irregular codes and their A-method modified versions, $N \approx 2000$, $R \approx 0.85$, $\Lambda(x) = 0.5x^4 + 0.5x^5$, and the QSPA-FFT decoder.

the topological or the weight conditions.

According to condition 1 in Lemma 1, an unlabeled non-binary elementary AS is a binary elementary AS. Since each binary AS is formed by a collection of cycles in the Tanner graph, we first focus our analysis on a single cycle in the protograph G. The following lemma identifies the relationship between a cycle in protograph G and its corresponding cycle(s) in the binary lifted graph $\hat{G}$.

Lemma 1. *([44]) Consider a cycle C_p involving p distinct variable nodes in the protograph with edges $e_1, e_2, \ldots, e_{2p}$. After lifting, C_p results in z cycles of the same length in the binary lifted graph if and only if the lifting parameters d_{e_i}, $i \in \{1, \ldots, 2p\}$ associated with the edges involved in C_p satisfy the following condition:*

$$\sum_{i=1}^{p} d_{e_{2i-1}} = \sum_{i=1}^{p} d_{e_{2i}} \mod z. \tag{3.7}$$

Otherwise, C_p results in one or more cycles of larger lengths.

Corollary 1. *Consider an (a, b) binary AS $A_{a,b}$ in the protograph G. After lifting G by the factor z to produce $\hat{G}$, $A_{a,b}$ results in z ASs of the same size in $\hat{G}$ if for every cycle C_p of length $2p$ in $A_{a,b}$, the lifting parameters d_{e_i}, $i \in \{1, 2, \ldots, 2p\}$ associated with edges of C_p satisfy* (3.7)*.*

Based on Corollary 1, we can prevent the existence of absorbing sets in $\hat{G}$ by ensuring that for at least one cycle C_p of AS $A_{a,b}$ in G, the lifting parameters d_{e_i} do not satisfy (3.7).

The weight condition in Lemma 1 implies that a non-binary elementary AS not only satisfies the topological condition, i.e., the unlabeled subgraph is a binary elementary AS, but also the edge weights in all of its cycles satisfy (3.5).

We study how the weight condition of non-binary elementary ASs maps to the NB-QC code construction. To analyze the weight condition, we first consider a single cycle.

Lemma 2. *([53]) Consider that a cycle C_p in protograph G with edges $\{e_1, e_2, \ldots, e_{2p}\}$ results in z cycles of the same length in the binary lifted graph $\hat{G}$. The z copies of C_p*

satisfy the weight condition in (3.5) *if the labeling parameters* ρ_{e_i}, $i \in \{1, \ldots, 2p\}$ *associated with the edges involved in* C_p *satisfy the following condition:*

$$\sum_{i=1}^{p} \rho_{e_{2i-1}} = \sum_{i=1}^{p} \rho_{e_{2i}} \mod q-1. \tag{3.8}$$

Based on Lemma 1, the edge weights of all the cycles in a non-binary elementary AS satisfy the weight condition in (3.5). Note that an elementary AS typically consists of more than one cycle. Therefore, Lemma 2 implies the following corollary.

Corollary 2. *Consider that an elementary AS* $A_{a,b}$ *in the protograph* G *results in* z *binary elementary absorbing sets of the same size in* $\hat{G}$*. After edge weight assignment, the* z *copies of* $A_{a,b}$ *result in* z *non-binary elementary absorbing sets if for each cycle in* $A_{a,b}$*, the labeling parameters satisfy* (3.8)*.*

The above corollary offers an approach to avoid non-binary elementary ASs in the edge weight assignment step of the NB-QC code design. For a binary elementary AS present in the binary lifted graph $\hat{G}$, the labeling parameters should be chosen such that the weight condition is not satisfied for at least one cycle in the elementary absorbing set.

We now propose an algorithm to design NB-QC codes with an improved error floor performance. The main idea is to avoid non-binary elementary ASs in the Tanner graph of the designed NB-QC code. Based on our earlier discussion, an elementary AS $A_{a,b}$ in G results in z non-binary elementary ASs if the lifting and the labeling parameters associated with the edges in $A_{a,b}$ satisfy (3.7) and (3.8). In design approach, by informed selection of the lifting and the labeling parameters, we ensure that for each binary AS in G, at least one cycle does not satisfy either (3.7) or (3.8).

The inputs to the algorithm are: 1. A binary protograph G which determines the design rate, column weight and row weight of the code; 2. The finite field size, q, of the resulting NB-QC code; 3. The lifting factor, z.

The method is stated in Algorithm 1. We first construct a random NB-QC code

with random assignment of the lifting and labeling parameters to all the edges in the protograph G. Based on the parameters of G, we first choose the set W of pairwise parameters (i.e., $W = \{(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\}$ corresponding to the k elementary ASs which we wish to avoid[4].

We then find the smallest AS (a, b) in W and form the set U which includes all binary (a, b) ASs in protograph G. For each AS in U, we determine if all its cycles satisfy (3.7). If they do and if it is possible, we change the lifting parameters of the edges to ensure that at least one cycle in the AS does not satisfy (3.7). For each AS which is not avoided by the choice of lifting parameters, we determine if all the cycles satisfy (3.8) or not. If yes, the labeling parameter associated with an edge will be changed to ensure that at least one cycle of the AS does not satisfy (3.8). The new labeling parameter is chosen such that the previously canceled NB-ASs remain canceled. This process continues until either all ASs are canceled or no more ASs can be canceled.

Note that step 6 in Algorithm 1, where we find all (a, b) absorbing sets in the given protograph G, dominates the computational complexity of our proposed algorithm. Although it is proven that it is NP-complete to exhaustively find small error-prone substructures (stopping sets and trapping sets) in LDPC codes [54], several papers, such as [54] and [42], have proposed algorithms to reduce the computational complexity.

Note that our algorithm can be used to design both regular and irregular NB-QC codes. In the case of irregular NB-QC codes, the protograph G is an irregular Tanner graph.

We now present the results of our simulations for different NB-QC codes[5]. We report bit error rate (BER) figures to compare the performance of our designed codes with other state-of-the-art NB-QC codes. We also present the error profiles of the decoder for different code constructions which explain the superior performance of our

[4]Note that the parameters of G, such as the column weight and girth determine the ASs available in G. For example, $(4, 4)$ ASs are possible only when $c = 4$. For other choices of c, $(4, 4)$ ASs do not exist in G.

[5]We have performed additional simulations and have observed similar results for other choices of code parameters (blocklength, code rate, and column weight).

Algorithm 2 Design of NB-QC codes with reduced number of non-binary elementary ASs.

Inputs: Protograph G, field size q and lifting factor z.

Randomly assign a $d_e \in \{0, 1, \ldots, z-1\}$ and a $\rho_e \in \{0, 1, \ldots, q-2\}$ to each edge e in G.

Choose W, the set of all ASs to be canceled .

Let $C = \varnothing$ be the set of ASs which can not be eliminated in the lifting process.

for $\forall (a, b)$ AS $\in W$ **do**

    Find U, the set of all ASs of size (a, b) in G.

    for $\forall S \in U$ **do**

        Let F_S be the set of all the cycles in S.

        If at least one of the cycles in F_S does not satisfy (3.7), go to the next AS S in U.

        Let E_S be the list of all the edges involved in F_S.

        Find an edge e in E_S such that there exist a $d'_e \neq d_e$ which guarantees that at least one of the cycles in F_S does not satisfy (3.7). If the value exists, replace d_e with d'_e and go to 7, else $C \leftarrow C \cup S$.

    end for

end for

for $\forall$ absorbing set $S \in C$ **do**

    Let F_S be the set of cycles of S.

    If at least one of the cycles in F_S does not satisfy (3.8), go to next absorbing set S in C.

    Let E_S be the list of all the edges involved in F_S.

    Find edge e in E_S such that there exist a $\rho'_e \neq \rho_e$ which guarantees that at least one of the cycles in F_S does not satisfy (3.8). If the value exists, replace ρ_e with ρ'_e.

end for

designed codes. The following is the list of the code constructions that we consider:

1) **Random construction**: For the given protograph, we randomly assign the lifting and labeling parameters to each edge.

2) **ACE construction**: We compare our results with the method recently introduced in [44]. The algorithm in [44] has two steps. First, for each cycle in the given protograph G, the algorithm finds the ACE value, which is defined as $\sum_{v_i} d_{v_i} - 2$, where d_{v_i} is the degree of the node v_i, and the summation is over all the variable nodes of the cycle. Then, it searches for cycles in G which their associated ACE value is greater than a bound which is given as an input to the algorithm. Then the algorithm attempts to eliminate them in the lifted graph $\hat{G}$, by properly choosing the lifting parameters. In the second step, the cycles in $\hat{G}$ that violate a non-binary ACE constraint are found. Then, the algorithm attempts to cancel these cycles by carefully choosing the labeling parameters.

3) **Absorbing set (AS) construction**: Based on our proposed approach stated in Algorithm 1, we first identify a list of problematic ASs to cancel. The algorithm attempts to cancel these ASs by informed selection of the lifting and labeling parameters. Note that unlike the ACE approach, we are able to cancel more problematic ASs by canceling one cycle per AS in the Tanner graph of the code.

Figure 3.8 shows the simulation results for the three different constructions over field sizes $q = 4, 8, 16$, design rate $R = 0.69$, column weight $c = 4$ and row weight $\ell = 13$, transmitted over a binary-input additive white Gaussian noise (AWGN) channel. The set of curves over GF(4) has the following parameters: block length $N = 1014$ bits and lifting factor $z = 3$. The figure also includes the set of curves for GF(8) which have the following parameters: $N = 3549$ bits and lifting factor $z = 7$. For GF(16) set of curves, $N = 2028$ bits and lifting factor $z = 3$. Note that the ACE spectrum for the three ACE codes in Figure 3.8 is equal to $(\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \hat{\tau}_6^{(b)}, \hat{\tau}_8^{(b)}, \hat{\tau}_{10}^{(b)}) = (\infty, \infty, 6, 8, 10)$. Figure 3.8 shows that both ACE and AS approaches significantly improve the performance of the NB-QC code compared to the random approach. The

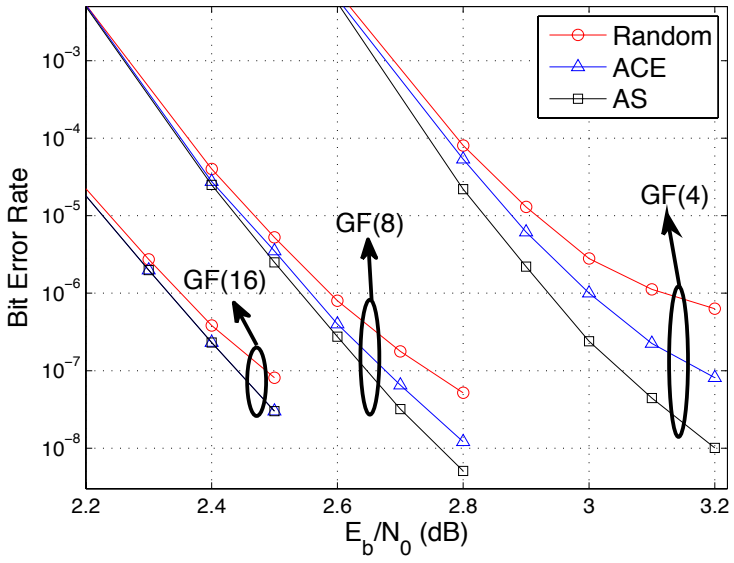


Figure 3.8: Performance comparison for NB-QC codes, with blocklength $N = 1014$ for codes over GF(4), $N = 3549$ for codes over GF(8) and $N = 2028$ for codes over GF(16), rate $R = 0.69$, and row weight $c = 4$.

AS approach achieves a better performance compared to the ACE approach since it focuses on canceling only one cycle per AS, whereas the ACE approach cancels all the cycles that violate the ACE constraints. Therefore, the AS approach is capable of removing more ASs. The performance comparison for different values of q and z reveals that the performance improvement for both ACE and AS approaches is more pronounced for smaller values of q and z, since for larger values of q and z, there are fewer number of non-binary absorbing sets to begin with. Table 3.3 includes the error profiles for GF(4) and GF(8) curves in Figure 3.8 at SNR$=$ 3.2dB and SNR$=$ 2.8dB, respectively. The table confirms that the AS approach cancels more problematic absorbing sets from the NB-QC code. The errors listed as 'other' in the table include 'oscillating' errors, in which the decoder oscillates between different errors in its last few iterations, and 'non-absorbing set' errors, in which the decoder converges to an error which is not an AS.

Figure 3.9 presents the simulation results for codes constructed by the three differ-

Table 3.3: Error Profile for the performance curves shown in Figure 3.8.

Error Type	$(6,4)$	$(6,6)$	$(7,4)$	$(8,2)$	$(8,4)$	$(9,4)$	$(10,4)$	other
GF(4), $N = 1014$ bits, SNR$= 3.2$ dB, $c = 4$								
Random	88	14	10	33	5	8	19	15
ACE	26	8	10	4	0	4	8	21
AS	0	0	0	0	0	0	0	23
GF(8), $N = 3549$ bits, SNR$= 2.8$ dB, $c = 4$								
Random	32	9	11	18	2	6	9	13
ACE	0	0	0	0	0	4	9	21
AS	0	0	0	0	0	0	0	29

Table 3.4: Error Profile for the performance curves shown in Figure 3.9.

Error Type	$(4,8)$	$(5,9)$	$(6,8)$	$(6,10)$	$(7,9)$	$(8,6)$	$(8,8)$	$(8,10)$	other
GF(4), $N = 726$ bits, SNR$= 3.3$ dB, $c = 5$									
Random	49	12	19	6	8	2	4	6	17
ACE	13	3	5	2	2	0	5	5	25
AS	0	0	0	0	0	0	0	0	31

ent approaches over field sizes $q = 4, 8$, design rate $R = 0.54$, column weight $c = 5$ and row weight $\ell = 11$. The constructed codes over GF(4) and GF(8) have $N = 726$ and $N = 2541$ bits, respectively. Note that the ACE spectrum for the three ACE codes in Figure 3.8 is equal to $(\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \hat{\tau}_6^{(b)}, \hat{\tau}_8^{(b)}, \hat{\tau}_{10}^{(b)}) = (\infty, \infty, 9, 12, 15)$. Similar to Figure 3.8, both the ACE approach and the AS approach have significantly better performance in the error floor region compared to the random construction of an NB-QC code. Table 3.4 confirms the superior performance of the AS approach: there exist fewer ASs in the error profile of the code constructed by the AS approach.

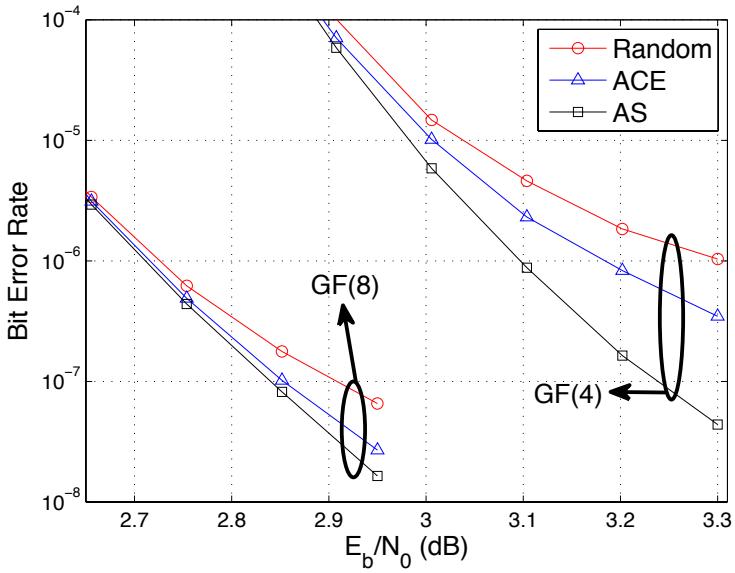


Figure 3.9: Performance comparison for NB-QC codes, blocklength $N = 726$ for codes over GF(4) and $N = 2541$ for codes over GF(8), $R = 0.54$, $c = 5$.

CHAPTER 4

Asymptotic Analysis of Binary and Non-Binary LDPC Code Ensembles

In our attempt to find non-binary graph-based codes suitable for emerging data storage systems, in this chapter, we study the asymptotic behavior of non-binary LDPC codes. Our asymptotic analysis provide a new toolbox for evaluating families of non-binary LDPC codes. For example, asymptotic distribution of objects of interest, such as codewords and absorbing sets, provides information about performance of ensembles of non-binary LDPC codes in error floor region.

The toolbox offered in this chapter can be used to determine whether a family of non-binary unstructured or structured LDPC codes are proper candidates for data storage applications.

4.1 Introduction

In this chapter, we consider two families of LDPC codes, regular unstructured LDPC codes and protograph-based LDPC codes. We provide asymptotic analysis and enumeration for objects of interest (codewords, psuedocodewords, trapping sets, and absorbing sets) in these two code families. The results presented in this chapter advance the analytical toolbox of non-binary graph-based codes.

- **Regular unstructured LDPC codes:** In Chapter 3, we showed that non-binary absorbing sets offer a more refined finite block length characterization of errors

for non-binary codes compared to trapping sets. Our results in this part extends this observation to the asymptotic regime. In particular, we first compute the normalized logarithmic asymptotic distributions of binary absorbing sets and fully absorbing sets, including elementary (fully) absorbing sets. The calculations are based on the trapping set enumeration method proposed by Milenkovic, Soljanin, and Whiting. We compare distributions of absorbing and trapping sets for representative code parameters of interest, and quantify the (lack of) discrepancies between the two. Good absorbing set properties are implied for known structured LDPC codes, including repeat accumulate codes and protograph-based constructions. Establishing the distribution of fully absorbing sets (especially when the discrepancy with the trapping set distribution is significant) allows one to further refine the estimates of the error rates under bit-flipping and related decoders.

We also find the asymptotic distribution of non-binary elementary absorbing sets in regular code ensembles by using techniques from graph theory. We also show that in the non-binary regime, as the alphabet size q gets larger, it is harder to satisfy edge labeling conditions of non-binary absorbing sets.

- **Non-binary protograph-based LDPC codes:** In this part, we generalize existing definitions and techniques from the binary to the non-binary domain and provide ensemble performance evaluation of the resultant non-binary protograph-based (NBPB) codes through the explicit computation of codeword, trapping sets and pseudocodewords enumerators.

4.2 Asymptotic Distribution of Absorbing Sets for Regular Binary and Non-Binary Unstructured LDPC Ensembles

The main contribution of this section is to establish the asymptotic analysis of binary and non-binary absorbing sets for regular LDPC code ensembles, and to, for the first

time, quantify the difference between the asymptotic distributions of absorbing sets and trapping sets.

4.2.1 Random Matrix Enumeration

An important tool for the code performance analysis is the enumeration of a random code ensemble with certain properties. It is convenient to introduce the following collection of matrices.

Definition 1. *Let $\Lambda_{n,m}^{\boldsymbol{l},\boldsymbol{r}}$ be the set of all $n \times m$ binary matrices with row-weight vector $\boldsymbol{l} = (l_1, l_2, ..., l_n)$ and column-weight vector $\boldsymbol{r} = (r_1, r_2, ..., r_m)$, where l_i, $1 \leq i \leq n$, represents the weight of the i^{th} row, and r_j, $1 \leq j \leq m$, represents the weight of the j^{th} column of matrices in $\Lambda_{n,m}^{\boldsymbol{l},\boldsymbol{r}}$.*

We use the following theorem from [55] to asymptotically enumerate parity check matrices under certain row and column weight constraints. Related approaches for analyzing asymptotic properties of LDPC codes are discussed in [56] and [17].

Theorem 2. *(cf. [55]) For given $\boldsymbol{l} = (l_1, l_2, ..., l_n)$ and $\boldsymbol{r} = (r_1, r_2, ..., r_m)$, let $t_{n,m}(\boldsymbol{l}, \boldsymbol{r}) = \left|\Lambda_{n,m}^{\boldsymbol{l},\boldsymbol{r}}\right|$ be the cardinality of the collection $\Lambda_{n,m}^{\boldsymbol{l},\boldsymbol{r}}$, and let $f = \sum_{i=1}^{n} l_i = \sum_{j=1}^{m} r_j$. For bounded values of l_i's and r_j's and constant ratio m/n, as $n \to \infty$,*

$$t_{n,m}(\boldsymbol{l}, \boldsymbol{r}) \doteq \frac{f!}{\prod_{i=1}^{n} l_i! \prod_{j=1}^{m} r_j!} \cdot \exp\left[-\frac{1}{2f^2} \sum_{i=1}^{n} l_i(l_i - 1) \sum_{j=1}^{m} r_j(r_j - 1)\right], \tag{4.1}$$

where the notation "$t_{n,m}(\boldsymbol{l}, \boldsymbol{r}) \doteq v_{n,m}(\boldsymbol{l}, \boldsymbol{r})$" stands for $\limsup_{n\to\infty} \left|\frac{t_{n,m}(\boldsymbol{l},\boldsymbol{r})}{v_{n,m}(\boldsymbol{l},\boldsymbol{r})} - 1\right| = 0$.

Lemma 3 shows how to enumerate a larger matrix by breaking it up into two matrices and enumerating them separately. First, we define three collections of auxiliary matrices. Given positive integers l, r, m, n_u and n, with $n_u < n$, and an integer-valued vector $\boldsymbol{r_u} = (r_1, ..., r_m)$ with each entry r_j satisfying $0 \leq r_j \leq r$, we let

- $\Lambda_{n,m}^{\bar{l},\bar{r}}$ be the set of $n \times m$ binary matrices with rows weighted l and columns weighted r (cf. Definition 1),

- $\Lambda_{n_u,m}^{\bar{l},\boldsymbol{r_u}}$ be the set of $n_u \times m$ binary matrices with rows weighted l and columns weighted $\boldsymbol{r_u} = (r_1, ..., r_m)$, and

- $\Lambda_{n_d,m}^{\bar{l},\boldsymbol{r_d}}$ be the set of $n_d \times m$ matrices, where $n_u + n_d = n$, with rows weighted l and columns with weights $\boldsymbol{r_d} = ((r - r_1), ..., (r - r_m))$.

As a shorthand, $\bar{\boldsymbol{l}}$ ($\bar{\boldsymbol{r}}$) denotes a vector with all entries equal to l (r) and with dimension n (m).

Lemma 3. *Let $\Lambda_{n,m}^{\bar{l},\bar{r}}$, $\Lambda_{n_u,m}^{\bar{l},\boldsymbol{r_u}}$ and $\Lambda_{n_d,m}^{\bar{l},\boldsymbol{r_d}}$ be the three sets of matrices as defined above. The cardinality of the set $\Lambda_{n,m}^{\bar{l},\bar{r}}$ is expressed in terms of cardinalities of constituent sets as,*

$$\left|\Lambda_{n,m}^{\bar{l},\bar{r}}\right| = \sum_{\{\boldsymbol{r_u}:\, 1\leq j\leq m,\, 0\leq r_j\leq r\}} \left|\Lambda_{n_u,m}^{\bar{l},\boldsymbol{r_u}}\right| \cdot \left|\Lambda_{n_d,m}^{\bar{l},\boldsymbol{r_d}}\right|. \tag{4.2}$$

Proof. For a particular choice of the vector $\boldsymbol{r_u}$, let $M_u \in \Lambda_{n_u,m}^{\bar{l},\boldsymbol{r_u}}$ and $M_d \in \Lambda_{n_d,m}^{\bar{l},\boldsymbol{r_d}}$. By pairing up M_u and M_d, such that $M = \begin{bmatrix} M_u \\ M_d \end{bmatrix}$, we obtain a regular matrix with rows weighted l and columns weighted r. There are $\left|\Lambda_{n_u,m}^{\bar{l},\boldsymbol{r_u}}\right| \cdot \left|\Lambda_{n_d,m}^{\bar{l},\boldsymbol{r_d}}\right|$ different regular matrices for a particular vector $\boldsymbol{r_u}$. Summing over all possible choices of $\boldsymbol{r_u}$ leads the overall count. □

For convenience, we let $G_{n,m}^{\boldsymbol{l},\boldsymbol{r}}$ be the set of all Tanner graphs corresponding to the parity check matrices whose transposes are in the $\Lambda_{n,m}^{\bar{l},\bar{r}}$ collection.

4.2.2 Asymptotic Distribution of Binary Absorbing Sets

In this section, we provide the normalized logarithmic asymptotic distribution of absorbing sets and fully absorbing sets for (l, r) regular LDPC code ensembles, where l and r denote the bit node and the check node degree, respectively. Furthermore, we

derive simplified formulas for the normalized logarithmic asymptotic distribution of the elementary (fully) absorbing sets for small values of l.

The normalized logarithmic asymptotic distribution of (a, b) absorbing sets is defined as

$$e^{l,r}(\theta, \lambda) \triangleq \lim_{n\to\infty} \frac{1}{n} \log p^{l,r}_{a,b,n} = \lim_{n\to\infty} \frac{1}{n} \log \frac{z^{l,r}_{a,b,n}}{\left|\Lambda^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}_{n,m}\right|}, \tag{4.3}$$

where $\theta = \frac{a}{n}$, $\lambda = \frac{b}{n}$ and $p^{l,r}_{a,b,n}$ is the average number of size (a, b) absorbing sets in a Tanner graph in $G^{\boldsymbol{l},\boldsymbol{r}}_{n,m}$. Here, $z^{l,r}_{a,b,n}$ is the number of (a, b) absorbing sets over all Tanner graphs in $G^{\boldsymbol{l},\boldsymbol{r}}_{n,m}$. Here and in the subsequent exposition, $\log$ is taken with base e.

We likewise define the normalized logarithmic asymptotic distribution of (a, b) fully absorbing sets as

$$e^{(f)^{l,r}}(\theta, \lambda) \triangleq \lim_{n\to\infty} \frac{1}{n} \log p^{(f)^{l,r}}_{a,b,n} = \lim_{n\to\infty} \frac{1}{n} \log \frac{z^{(f)^{l,r}}_{a,b,n}}{\left|\Lambda^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}_{n,m}\right|}, \tag{4.4}$$

where $p^{(f)^{l,r}}_{a,b,n}$ is the average number and $z^{(f)^{l,r}}_{a,b,n}$ is the total number of (a, b) fully absorbing sets in $G^{\boldsymbol{l},\boldsymbol{r}}_{n,m}$.

The normalized logarithmic asymptotic distributions of *elementary* (fully) absorbing sets are defined analogously to (4.24) and (4.4); we shall use the subscript E to denote the elementary attribute.

Let us consider an (a, b) absorbing set $A_{a,b}$ in a Tanner graph in the collection $G^{\boldsymbol{l},\boldsymbol{r}}_{n,m}$. For $1 \le i \le a$, let σ_i denote the number of edges that connect the i^{th} bit node in $A_{a,b}$ to satisfied check nodes, and for $1 \le j \le m$, let δ_j denote the number of edges that connect the j^{th} check node to the bit nodes in $A_{a,b}$. Theorem 3 establishes the asymptotic logarithmic scaling of the average number of (a, b) absorbing sets in $G^{\boldsymbol{l},\boldsymbol{r}}_{n,m}$. Here, "$p^{l,r}_{a,b,n} \cong w^{l,r}_{a,b,n}$" means "$\lim_{n\to\infty} \frac{1}{n} \log p^{l,r}_{a,b,n} = \lim_{n\to\infty} \frac{1}{n} \log w^{l,r}_{a,b,n}$".

Theorem 3. *Let* $0 < \zeta, \theta, \lambda < 1$ *where* $\zeta \triangleq \frac{m}{n} = \frac{l}{r}$. *Then,*

$$p^{l,r}_{a,b,n} \cong \sum_{\{\delta_j,\sigma_i:1\le j\le m,1\le i\le n,S\}} \frac{\binom{n}{a}\binom{m}{b}}{\binom{nl}{al}\binom{al}{q}} \prod_{i=1}^{a} \binom{l}{\sigma_i} \prod_{j=1}^{m} \binom{r}{\delta_j}, \tag{4.5}$$

where $q = \sum_{j=1}^{m-b} \delta_j$ *and the summation goes over all* δ_j*'s, and* σ_i*'s which satisfy the following set of conditions:*

$$S = \begin{cases} \sum_{j=1}^{m} \delta_j = al; \sum_{j=1}^{m-b} \delta_j = \sum_{i=1}^{a} \sigma_i; \\ \frac{l}{2} < \sigma_i \leq l \ \text{ for } 1 \leq i \leq a; \\ \delta_j \ \text{ is even for } 1 \leq j \leq m-b; \\ \delta_j \ \text{ is odd for } m-b+1 \leq j \leq m. \end{cases} \tag{4.6}$$

Proof. First, for the selected $A_{a,b}$ absorbing set we express the $m \times n$ parity check matrix H as follows (for convenience, we work with the transpose of matrix H):

$$H^T = \left[\begin{array}{c|c} M_1 & M_2 \\ \hline \multicolumn{2}{c}{M_3} \end{array}\right],$$

where M_1 is a size $a \times (m-b)$ binary matrix corresponding to the subgraph of the Tanner graph spanned by the bit nodes in $A_{a,b}$ and the check nodes that are connected to $A_{a,b}$ even number of times (including zero times). Therefore, the matrix M_1 only has even-weighted columns. The matrix M_2 is a size $a \times b$ binary matrix corresponding to subgraph of the Tanner graph spanned by the bit nodes in $A_{a,b}$ and the check nodes that are connected to $A_{a,b}$ odd number of times. The matrix M_2 only has odd-weighted columns. The matrix M_3 is an $(n-a) \times m$ binary matrix corresponding to the remainder of the Tanner graph.

Then, for $1 \leq j \leq m$, δ_j is the weight of the j^{th} column of the submatrix $[M_1|M_2]$, and for $1 \leq i \leq a$, σ_i is the weight of the i^{th} row of the submatrix M_1. To ensure that the row weight across H^T is l, the weight of the i^{th} row in M_2 is $(l - \sigma_i)$.

Likewise, the weights of columns of M_3 are chosen with respect to the column weights of matrices M_1 and M_2 to ensure that all columns in H^T are weighted r: the j^{th} column of M_3 has weight $(r - \delta_j)$. Also, by definition of an absorbing set, a size (a, b) absorbing set requires $\sigma_i > \frac{l}{2}$ for all $1 \leq i \leq a$.

Let $\Lambda_{v,v'}$ refer to the set of all $v \times v'$ binary matrices. Given non-negative integers a, b, m, n, l, r with $a \leq n$, $b \leq m$ and $nl = mr$, and given non-negative integer valued

vectors $(\delta_1, \ldots, \delta_m)$ and $(\sigma_1, \ldots, \sigma_a)$ with $0 \leq \delta_j \leq r$, $\forall j$, and with $0 \leq \sigma_i \leq l$, $\forall i$, let us define the following collections of matrices:

$$\begin{aligned}
\Lambda_1 =&\{\forall M \in \Lambda_{a,m-b} : \quad \text{for } 1 \leq i \leq a, \sum_{h=1}^{m-b} M(i,h) = \sigma_i; \\
&\text{for } 1 \leq j \leq m-b, \sum_{g=1}^{a} M(g,j) = \delta_j\}, \\
\Lambda_2 =&\{\forall M \in \Lambda_{a,b} : \text{for } 1 \leq i \leq a, \sum_{h=1}^{b} M(i,h) = l - \sigma_i; \\
&\text{for } 1 \leq j \leq b, \sum_{g=1}^{a} M(g,j) = \delta_{m-b+j}\}, \\
\Lambda_3 =&\{\forall M \in \Lambda_{n-a,m} : \text{for } 1 \leq i \leq n-a, \sum_{h=1}^{m} M(i,h) = l; \\
&\text{for } 1 \leq j \leq m, \sum_{g=1}^{n-a} M(g,j) = r - \delta_j\}.
\end{aligned}$$

The asymptotic cardinalities of sets Λ_1, Λ_2, Λ_3 and $\Lambda_{n,m}^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}$ are computed using Theorem 2, and are shown in (4.7) through (4.10).

The matrix H^T is an $n \times m$ matrix with all rows weighted l and all columns weighted r which has fixed row and column orderings fixed by the choice of M_1, M_2 and M_3. Using Lemma 3 and accounting for the choice of which a out of n bit nodes (b out of m check nodes) constitute the absorbing set (odd degree neighbors to the absorbing set) we obtain:

$$z_{a,b,n}^{l,r} = \sum_{\{\delta_j, \sigma_i : 1 \leq j \leq m, 1 \leq i \leq n, S\}} \binom{n}{a} \binom{m}{b} |\Lambda_1| \, |\Lambda_2| \, |\Lambda_3| \ , \tag{4.11}$$

where the condition set S is given by (4.6). Therefore, the average number of size (a, b)

$$|\Lambda_1| \doteq \frac{\left(\sum_{j=1}^{m-b} \delta_j\right)!}{\prod_{i=1}^{a} \sigma_i! \prod_{j=1}^{m-b} \delta_j!} \cdot \exp\left[-\frac{\sum_{i=1}^{a} \sigma_i(\sigma_i - 1) \sum_{j=1}^{m-b} \delta_j(\delta_j - 1)}{2\left(\sum_{j=1}^{m-b} \delta_j\right)^2} \right], \tag{4.7}$$

$$|\Lambda_2| \doteq \frac{\left(\sum_{j=m-b+1}^{m} \delta_j\right)!}{\prod_{i=1}^{a} (l - \sigma_i)! \prod_{j=m-b+1}^{m} \delta_j!} \cdot \exp\left[-\frac{\sum_{i=1}^{a} (l - \sigma_i)(l - \sigma_i - 1) \sum_{j=m-b+1}^{m} \delta_j(\delta_j - 1)}{2\left(\sum_{j=m-b+1}^{m} \delta_j\right)^2} \right], \tag{4.8}$$

$$|\Lambda_3| \doteq \frac{((n-a)l)!}{(l!)^{n-a} \prod_{j=1}^{m} (r - \delta_j)!} \cdot \exp\left[-\frac{(n-a)l(l-1) \sum_{j=1}^{m} ((r - \delta_j)(r - \delta_j - 1)).}{2((n-a)l)^2} \right], \tag{4.9}$$

$$\left|\Lambda_{n,m}^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}\right| \doteq \frac{(nl)!}{(l!)^n (r!)^m} \cdot \exp\left[-\frac{nl(l-1)\, mr(r-1)}{2(nl)^2} \right]. \tag{4.10}$$

absorbing sets in a Tanner graph in $G_{n,m}^{\boldsymbol{l},\boldsymbol{r}}$ is

$$\begin{aligned} p_{a,b,n}^{l,r} =& \frac{z_{a,b,n}^{l,r}}{\left|\Lambda_{n,m}^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}\right|} \\ =& \frac{\sum_{\{\delta_j, \sigma_i : 1 \le j \le m, 1 \le i \le n, S\}} \binom{n}{a} \binom{m}{b} |\Lambda_1|\, |\Lambda_2|\, |\Lambda_3|}{\left|\Lambda_{n,m}^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}\right|}. \end{aligned} \tag{4.12}$$

Note that the exponential terms in (4.7) through (4.10) become negligible under the $\frac{1}{n}\log(\cdot)$ operation in the limit $n \to \infty$, since both the numerator and denominator of each of the terms within the exponents grow quadratic with n. For example, in (4.7), in the numerator of the term within the exponent, $\sum_{i=1}^{a} \sigma_i(\sigma_i - 1)$ is a summation of

$a = \theta n$ bounded terms, and $\sum_{j=1}^{m-b} \delta_j(\delta_j - 1)$, is also a summation of $m - b = (\zeta - \theta)n$ bounded terms. Each summation grows linearly with n. Since the term $\left(\sum_{j=1}^{m-b} \delta_j\right)^2$ in the denominator grows quadratic with n, the ratio vanishes under the $\frac{1}{n}\log(\cdot)$ operation. With some manipulations, it readily follows that

$$p_{a,b,n}^{l,r} \cong \sum_{\{\delta_j,\sigma_i:1\leq j\leq m,1\leq i\leq n,S\}} \frac{\binom{n}{a}\binom{m}{b}}{\binom{nl}{al}\binom{al}{q}} \prod_{i=1}^{a} \binom{l}{\sigma_i} \prod_{j=1}^{m} \binom{r}{\delta_j}.$$

□

It is worth pointing out that the presented enumeration method does not double count the configurations of interest since distinct matrices are counted exactly once. However, certain configurations, by virtue of node symmetry, may be more likely.

For small trapping sets, most of the check nodes incident to the bit nodes in the trapping set connect to the trapping set at most twice, and are therefore elementary trapping sets [17, 57]. It is therefore of interest to also quantify distributions of elementary absorbing sets in this regime.

Corollary 3. *The normalized logarithmic asymptotic distribution of* $(a = \theta n, b = \lambda n)$ *elementary absorbing sets in* $G_{n,m}^{\mathbf{3,r}}$*, is given by (recall that "E" stands for "elementary"):*

$$\begin{aligned} e_E^{3,r}(\theta,\lambda) &= -2H_b(\theta, 1-\theta) - H_b(\zeta, 1-\zeta) \\ &\quad + \frac{3\theta-\lambda}{2}\log\binom{r}{2} + \theta H_b\left(\frac{\lambda}{\theta}, 1-\frac{\lambda}{\theta}\right) \\ &\quad - 3\theta H_b\left(\frac{\lambda}{3\theta}, 1-\frac{\lambda}{3\theta}\right) + \lambda\log 3r \\ &\quad + H_b\left(1-\zeta, \lambda, \frac{3\theta-\lambda}{2}, \frac{2\zeta-3\theta-\lambda}{2}\right), \end{aligned} \tag{4.13}$$

where $\zeta = \frac{m}{n}$*, and* $H_b(p_1, ..., p_N) = -\sum_{i=1}^{N} p_i \log p_i$*, with* $\sum_{i=1}^{N} p_i = 1$*, denotes the entropy function.*

Proof. In an elementary absorbing set, check nodes are connected to bit nodes in the absorbing set at most twice. Therefore, δ_j's have values 0, 1 or 2. For $m - b + 1 \leqslant j \leqslant$

m, $\delta_j = 1$ since each unsatisfied check has only one edge connected to the absorbing set.

Following the constraints on values of σ_i's (for $1 \leq i \leq a$, $\frac{l}{2} < \sigma_i \leq l$), σ_i's are either 2 or 3 for $1 \leq i \leq a$. Since each column of M_2 has weight 1, the total weight of matrix M_2 is λn. Each row of M_2 is weighted 1 or 0, and λn number of rows of M_2 are weighted 1. Thus, λn number of σ_i's are equal to 2 and the rest of σ_i's are equal to 3. Since the total weight in $[M_1|M_2]$ is $3\theta n$, it follows that $\frac{(3\theta-\lambda)n}{2}$ number of δ_j's are equal to 2 and rest of them are equal to 0. By substituting in the numerical values for δ_j's and σ_i's in (4.5), we have:

$$e_E^{3,r}(\theta,\lambda) = \frac{1}{n}\log \sum_{\{\delta_j,\sigma_i:1\leq j\leq m,1\leq i\leq n,S\}} \frac{\binom{n}{\theta n}\binom{\zeta n}{\lambda n}}{\binom{3\theta n}{(3\theta-\lambda)n}\binom{3n}{3\theta n}} \binom{3}{2}^{\lambda n} \binom{3}{3}^{(\theta-\lambda)n} \binom{r}{0}^{\frac{(2\zeta-\lambda-3\theta)n}{2}} \binom{r}{1}^{\lambda n} \binom{r}{2}^{\frac{(3\theta-\lambda)n}{2}}. \tag{4.14}$$

Note that $\frac{(3\theta-\lambda)n}{2}$ of even-valued δ_i's need to be 2 and λn of σ_j's also need to be 2. The summation reduces to multiplying the summand (which is now the same for all terms), by the number of ways the values of δ_i's and σ_j's can be selected. The total number of ways is $\binom{(\zeta-\lambda)n}{\frac{(3\theta-\lambda)n}{2}}\binom{\theta n}{\lambda n}$. For $n \to \infty$, we simplify (4.14) using Stirling's approximation, $\log(n!) \approx n\log n - n$, and the binomial approximation, $\log\binom{n}{\rho n} \approx nH_b(\rho, 1-\rho)$, to obtain the result. □

A special case of Corollary 3 is when $\lambda << \theta$, i.e., when there exists only a small fraction of unsatisfied checks. In this case, we can approximate (4.13) as:

$$\begin{aligned} &-2H_b(\theta,1-\theta) + \frac{3\theta}{2}\log\binom{r}{2} - H_b(\zeta,1-\zeta) \\ &+ H_b\left(1-\zeta, \frac{3\theta}{2}, \frac{2\zeta-3\theta}{2}\right), \end{aligned} \tag{4.15}$$

which is exactly the same result as in *Corollary 3.1* of [17] for the enumeration of elementary trapping sets. Therefore when $\lambda << \theta$ and θ is small, most (elementary) trapping sets satisfy the (elementary) absorbing set conditions. This observation is also shown in Fig. 4.6.

We derive an analogous result for when the bit node degree is 4.

Corollary 4. *The normalized logarithmic asymptotic distribution of* $(a = \theta n, b = \lambda n)$ *elementary absorbing sets in* $G^{4,r}_{n,m}$ *is given by:*

$$\begin{aligned} e_E^{4,r}(\theta,\lambda) &= -3H_b(\theta, 1-\theta) + \lambda \log 4r \\ &\quad + \frac{4\theta-\lambda}{2}\log\binom{r}{2} + \theta H_b\left(\frac{\lambda}{\theta}, 1-\frac{\lambda}{\theta}\right) \\ &\quad - 4\theta H_b\left(\frac{\lambda}{4\theta}, 1-\frac{\lambda}{4\theta}\right) - H_b(\zeta, 1-\zeta) \\ &\quad + H_b\left(1-\zeta, \lambda, \frac{4\theta-\lambda}{2}, \frac{2\zeta-4\theta-\lambda}{2}\right). \end{aligned} \tag{4.16}$$

Proof. Similarly to the proof of Corollary 3, δ_j's again have values 0, 1 or 2, and for $m-b+1 \leq j \leq m$, $\delta_j = 1$. Now, σ_i is either 3 or 4, and there are λn rows in M_2 corresponding to $\sigma_i = 3$. The rest of the proof mimics that of Corollary 3. □

In the case of fully absorbing sets, every bit node, irrespective of whether it belongs to the particular absorbing set or not, has fewer edges connected to the unsatisfied checks than other checks (checks being unsatisfied with respect to the absorbing set). Theorem 4 considers this additional constraint and provides the asymptotic scaling of the average number of fully absorbing sets. Recall that for $1 \leq j \leq n$, δ_j denotes the number of edges that connect the j^{th} check node to the bit nodes in $A^{(f)}_{a,b}$, and for $1 \leq i \leq a$, σ_i denotes the number of edges that connect the i^{th} bit node in the fully absorbing set $A^{(f)}_{a,b}$ to satisfied check nodes. Additionally, let μ_k, for $1 \leq k \leq n-a$, be the number of edges that connect the k^{th} bit node from the subset of bit nodes not in the fully absorbing set (that is, for $1 \leq k \leq n-a$) to the $(m-b)$ check nodes that themselves have even number of connections (including zero) to $A^{(f)}_{a,b}$.

Theorem 4. *Let $0 < \zeta, \theta, \lambda < 1$ where $\zeta = \frac{m}{n} = \frac{l}{r}$. Then, as $n \to \infty$, ${p^{(f)}}^{l,r}_{a,b,n}$, the average number of $(a = \theta n, b = \lambda n)$ fully absorbing sets in $G^{\boldsymbol{l,r}}_{n,m}$ scales as,*

$$
\begin{aligned}
{p^{(f)}}^{l,r}_{a,b,n} \cong & \sum_{\{\delta_j, \sigma_i, \mu_k : 1 \le j \le m, 1 \le i \le a, 1 \le k \le n-a, S, S_1\}} \frac{\binom{n}{a}\binom{m}{b}}{\binom{nl}{al}\binom{al}{q}} \\
& \frac{1}{\binom{(n-a)\cdot l}{(m-b)\cdot r - q}} \cdot \prod_{i=1}^{a} \binom{l}{\sigma_i} \prod_{k=1}^{n-a} \binom{l}{\mu_k} \prod_{j=1}^{m} \binom{r}{\delta_j},
\end{aligned} \tag{4.17}
$$

where $q = \sum_{j=1}^{m-b} \delta_j$. The summation goes over all δ_j's, σ_i's, and μ_k's under the conditions S given in (6) and S_1, where

$$
S_1 = \begin{cases} \sum_{k=1}^{n-a} \mu_k = (m-b)r - q, \\ \forall k \in \{1, 2, ..., n-a\}, \frac{l}{2} < \mu_k \le l. \end{cases} \tag{4.18}
$$

Proof. The proof of the theorem extends the proof of Theorem 3. For a given (a, b) fully absorbing set $A^{(f)}_{a,b}$, express the transpose of the $m \times n$ parity check matrix H as:

$$
H^T = \left[\begin{array}{c|c} M_1 & M_2 \\ \hline M_{31} & M_{32} \end{array} \right],
$$

where M_1 and M_2 are defined as before: the binary matrix M_1 of size $a \times (m-b)$ (the binary matrix M_2 of size $a \times b$), corresponds to the subgraph spanned by the bit nodes in $A^{(f)}_{a,b}$ and the check nodes that are connected to $A^{(f)}_{a,b}$ even (odd) number of times. The matrix M_{31} is an $(n-a) \times (m-b)$ binary matrix corresponding to the subgraph of the Tanner graph spanned by $(n-a)$ bit nodes not in $A^{(f)}_{a,b}$ and $(m-b)$ check nodes that are connected to the bit nodes in $A^{(f)}_{a,b}$ even number of times. Likewise, the matrix M_{32} is an $(n-a) \times b$ matrix that corresponds to the subgraph spanned by $(n-a)$ bit nodes not in $A^{(f)}_{a,b}$ and b check nodes that are themselves connected to the bit nodes in $A^{(f)}_{a,b}$ odd number of times. Given the fully absorbing set constraints, $\frac{l}{2} < \mu_k \le l$ for $1 \le k \le n-a$. The values of μ_k's are also constrained by the δ_j's.

Following the discussion in the proof of Theorem 3, the quantities $|\Lambda_1|$, $|\Lambda_2|$, and $\left|\Lambda^{\bar{\boldsymbol{l}},\bar{\boldsymbol{r}}}_{n,m}\right|$ remain the same as in equations (4.7), (4.8) and (4.10). The original matrix

$$|\Lambda_{31}| \doteq \frac{\left(\sum_{j=1}^{m-b}(r-\delta_j)\right)!}{\prod_{k=1}^{n-a}\mu_k! \prod_{j=1}^{m-b}(r-\delta_j)!} \cdot \exp\left[-\frac{\sum_{k=1}^{n-a}\mu_k(\mu_k-1)\sum_{j=1}^{m-b}(r-\delta_j)(r-\delta_j-1)}{2\left(\sum_{j=1}^{m-b}(r-\delta_j)\right)^2}\right], \tag{4.20}$$

$$|\Lambda_{32}| \doteq \frac{\left(\sum_{j=m-b+1}^{m}(r-\delta_j)\right)!}{\prod_{k=1}^{n-a}(l-\mu_k)! \prod_{j=m-b+1}^{m}(r-\delta_j)!} \cdot \exp\left[-\frac{\sum_{k=1}^{n-a}(l-\mu_k)(l-\mu_k-1)\sum_{j=m-b+1}^{m}(r-\delta_j)(r-\delta_j-1)}{2\left(\sum_{j=m-b+1}^{m}(r-\delta_j)\right)^2}\right]. \tag{4.21}$$

M_3 is partitioned into two new matrices, M_{31} and M_{32}. We define Λ_{31} and Λ_{32} matrix collections as:

$$\begin{aligned}
\Lambda_{31} =& \{\forall M \in \Lambda_{(n-a),(m-b)} : \text{for} 1 \leq k \leq n-a, \sum_{h=1}^{m-b} M(k,h) = \mu_k; \\
& \text{for } 1 \leq j \leq m-b, \sum_{g=1}^{n-a} M(g,j) = r - \delta_j\}, \\
\Lambda_{32} =& \{\forall M \in \Lambda_{n-a,b} : \text{for} 1 \leq k \leq n-a, \sum_{h=1}^{b} M(k,h) = l - \mu_k; \\
& \text{for} 1 \leq j \leq b, \sum_{g=1}^{n-a} M(g,j) = r - \delta_{m-b+j}\}.
\end{aligned}$$

The asymptotic scaling of $|\Lambda_{31}|$ and $|\Lambda_{32}|$ are given in (4.20) and (4.21). Now, by using Lemma 3, we have

$$\begin{aligned}
z^{(f)}{}_{a,b,n}^{l,r} =& \sum_{\{\delta_j,\sigma_i,\mu_k:1\leq j\leq m,1\leq i\leq a,1\leq k\leq n-a,S,S_1\}} \binom{n}{a}\binom{m}{b} \\
& \cdot |\Lambda_1|\,|\Lambda_2|\,|\Lambda_{31}|\,|\Lambda_{32}|\,.
\end{aligned} \tag{4.19}$$

Following the calculations in the proof of Theorem 2, as $n \to \infty$, we have

$$p^{(f)}{}^{l,r}_{a,b,n} \cong \sum_{\{\delta_j,\sigma_i,\mu_k:1\le j\le m,1\le i\le a,1\le k\le n-a,S,S_1\}} \frac{\binom{n}{a}\binom{m}{b}}{\binom{nl}{al}\binom{al}{q}} \cdot \frac{1}{\binom{(n-a)\cdot l}{(m-b)\cdot r-q}} \cdot \prod_{i=1}^{a}\binom{l}{\sigma_i}\prod_{k=1}^{n-a}\binom{l}{\mu_k}\prod_{j=1}^{m}\binom{r}{\delta_j}.$$

□

Based on Theorem 4, we now compute the normalized logarithmic asymptotic distribution of elementary fully absorbing sets for bit degrees 3 and 4.

Corollary 5. *The normalized logarithmic asymptotic distribution of* $(a = \theta n, b = \lambda n)$ *elementary fully absorbing set in* $G^{3,r}_{n,m}$ *is given by:*

$$\begin{aligned} e_E^{(f)}{}^{3,r}(\theta,\lambda) = & -2H_b(\theta,1-\theta) + \lambda\log 3r - H_b(\zeta,1-\zeta) \\ & + \frac{3\theta-\lambda}{2}\log\binom{r}{2} + \theta H_b(\frac{\lambda}{\theta},1-\frac{\lambda}{\theta}) \\ & - 3\theta H_b(\frac{\lambda}{3\theta},1-\frac{\lambda}{3\theta}) \\ & + H_b(1-\zeta,\lambda,\frac{3\theta-\lambda}{2},\frac{2\zeta-3\theta-\lambda}{2}) \\ & + [(1-\theta)H_b(\frac{\lambda(r-1)}{1-\theta},1-\frac{\lambda(r-1)}{1-\theta}). \\ & - 3(1-\theta)H_b.(\frac{\lambda(r-1)}{3(1-\theta)},1-\frac{\lambda(r-1)}{3(1-\theta)}) \\ & + \lambda(r-1)\log 3\,]. \end{aligned} \tag{4.22}$$

Proof. Here, the constraints over δ_j's and σ_i's are the same as in Corollary 3. Each bit node in the Tanner graph has fewer connections to unsatisfied checks than to satisfied checks, irrespective of whether or not it belongs to the fully absorbing set. Since each row of M_{32} has fewer than half of the 1's in each row of the overall matrix (here $l = 3$), each row of M_{32} has weight one or zero. In matrix M_{32}, because every column is weighted $r-1$, we have $(r-1)b = (r-1)\lambda n$ number of 1's. So, $(r-1)\lambda n$ number of rows of M_{32} are weighted 1 or equivalently $(r-1)\lambda n$ rows of M_{31} have a weight of two. Thus, $(r-1)\lambda n$ number of μ_k's are equal to 2 and $n(1-\theta) - (r-1)\lambda n$

of μ_k's are equal to 3. By plugging in δ_j's, σ_i's and λ_k's in Theorem 4 and using the approximations as in Corollary 3, the result follows. □

As $\lambda \to 0$, the equation (4.22) reduces to (4.15), where in this regime, (elementary) trapping sets, absorbing sets, and fully absorbing sets are approximately the same objects. This observation is illustrated in Fig. 4.1. The part inside the brackets of (4.22) comes from the additional constraints when an absorbing set is also a fully absorbing set. When θ and λ are small, we use $\log(x+1) \approx x$ for $x \to 0$, to simplify this additional summand. Then, the term inside the brackets is approximately equal to $-2\lambda^2(r-1)^2/3$. Note that the value of this expression is always negative and it is strictly decreasing as λ increases. It follows that the discrepancy between elementary absorbing sets and elementary fully absorbing sets increases as λ increases (shown in Fig. 4.1).

Using similar approximations, the following consequence of Theorem 4 readily follows.

Corollary 6. *The normalized logarithmic asymptotic distribution of* $(a = \theta n, b = \lambda n)$ *elementary fully absorbing set in* $G^{(4,r)}_{n,m}$ *is given by*

$$\begin{aligned} e_E^{(f)^{4,r}}(\theta,\lambda) &= -3H_b(\theta,1-\theta) + \lambda\log 4r \\ &\quad + \frac{4\theta-\lambda}{2}\log\binom{r}{2} + \theta H_b(\frac{\lambda}{\theta}, 1-\frac{\lambda}{\theta}) \\ &\quad - 4\theta H_b(\frac{\lambda}{4\theta}, 1-\frac{\lambda}{4\theta}) - H_b(\zeta, 1-\zeta) \\ &\quad + H_b(1-\zeta, \lambda, \frac{4\theta-\lambda}{2}, \frac{2\zeta-4\theta-\lambda}{2}) \\ &\quad + [(1-\theta)H_b(\frac{\lambda(r-1)}{1-\theta}, 1-\frac{\lambda(r-1)}{1-\theta}). \\ &\quad - 4(1-\theta)H_b.(\frac{\lambda(r-1)}{3(1-\theta)}, 1-\frac{\lambda(r-1)}{3(1-\theta)}) \\ &\quad + \lambda(r-1)\log 4\,]. \end{aligned} \tag{4.23}$$

Note that there exists a limit for the value of λ for absorbing sets to exist.

Remark 1. *Consider the (fully) absorbing sets in the Tanner graphs in* $G^{3,r}_{n,m}$. *Note that in order to satisfy the (fully) absorbing set condition, there can be at most one edge connecting a bit node in the absorbing set with an unsatisfied check. Therefore, the maximum number of edges connecting bit nodes from an absorbing set of size* $a = \theta n$ *with unsatisfied checks is* $a = \theta n$. *It follows that the maximum number of unsatisfied checks that can be induced by this absorbing set is also* $a = \theta n$, *so* $\frac{\lambda}{\theta} \leq 1$ *for* $b = \lambda n$. *In addition, since* $1 - R = \frac{l}{r} = \frac{m}{n}$, *where* $R, R < 1$, *is the design rate,* λ *is also upper bounded by* $1 - R$. *The same argument follows for* $G^{4,r}_{n,m}$.

This observation can be used to narrow down the range of parameters where the trapping sets are stable under bit-flipping decoding (e.g., in [57] trapping sets with $\frac{\lambda}{\theta} > 1$, and therefore unstable, were considered in the analysis).

Fig. 4.1 compares the normalized logarithmic asymptotic distributions of elementary trapping sets and elementary (fully) absorbing sets for different $G^{3,r}_{n,m}$, for $\theta = 0.001$. In this regime, elementary absorbing sets approximate absorbing sets well. As λ increases, the discrepancy between the trapping set and absorbing set distributions increases, because it becomes more difficult to meet additional (fully) absorbing set constraints. The discrepancy becomes more pronounced as the ratio $\zeta = \frac{l}{r}$ is lowered ($\Lambda^{3,6}_{n,m}$ vs. $\Lambda^{3,15}_{n,m}$ vs. $\Lambda^{3,30}_{n,m}$ collection), since having fewer checks of higher degrees makes it more difficult to meet additional combinatorial constraints.

Fig. 4.2 shows the comparison of the normalized logarithmic asymptotic distributions of elementary absorbing sets for $G^{3,r}_{n,m}$ and $G^{4,r}_{n,m}$, for θ set to 0.001. The results are reported for $\zeta = 0.5$ (for (l, r) equal to $(3, 6)$ and to $(4, 8)$) and for $\zeta = 0.25$ (for (l, r) equal to $(3, 15)$ and to $(4, 20)$). There are significantly fewer absorbing sets of a particular size in $G^{4,r}_{n,m}$ than in $G^{3,r}_{n,m}$ because in the former case additional edges in the graph make the combinatorial constraints of absorbing sets harder to meet. Also note that the curve for the $(4, 8)$ codes sits entirely below the horizontal line at 0 which means that in the asymptotic limit the number of absorbing sets is subexponential for these choices of θ and λ.

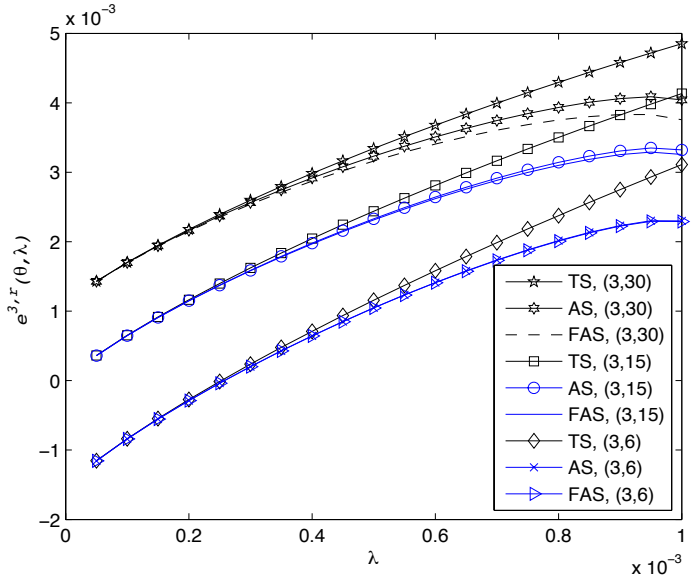


Figure 4.1: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) and elementary fully absorbing sets (FAS) for fixed $\theta = 0.001$ for various $G_{n,m}^{3,r}$ collections, each indexed by $(3, r)$. The equations in [17] are used to plot the trapping set curves. Note that for the $(3, 6)$ case the curves for absorbing sets and for fully absorbing sets are completely overlapping.

Fig. 4.3 and Fig. 4.4 show the discrepancy between the normalized logarithmic asymptotic distributions of elementary trapping sets and absorbing sets for $G_{n,m}^{3,6}$ and $G_{n,m}^{3,15}$, parameterized by the number of unsatisfied checks λ. From Remark 1, it follows that absorbing sets in $G_{n,m}^{3,6}$ and $G_{n,m}^{3,15}$ can exist only when $\frac{\lambda}{\theta} \leq 1$. Thus, the domain of θ is different for different λ's. Observe that in these two figures, for fixed θ, $e^{l,r}(\theta, \lambda)$ increases as λ increases for both trapping and absorbing sets. This agrees with the result from Fig. 4.1. We quickly remark that for $r < 9$ ($r \geq 9$), the curves are as those shown in Fig. 4.3 (Fig. 4.4). Thus for $r \geq 9$, there are exponentially many absorbing sets parametrized by given θ and λ, whereas for $r < 9$ there is a subexponential number of absorbing sets when θ and λ are sufficiently small. Figures 4.3 and 4.4 also show that as θ increases or as λ decreases, trapping sets become better proxies for absorbing sets.

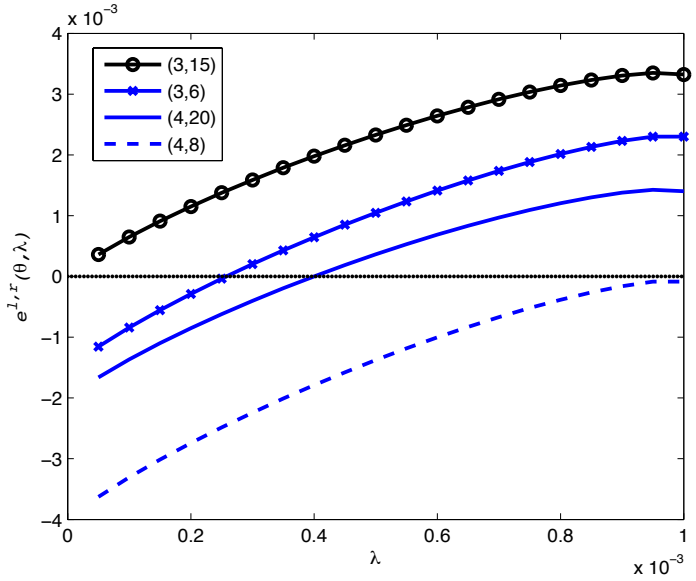


Figure 4.2: Comparison of the normalized logarithmic asymptotic distributions of elementary absorbing sets in $G_{n,m}^{\boldsymbol{3,r}}$ and $G_{n,m}^{\boldsymbol{4,r}}$ for fixed $\theta = 0.001$, and for ζ equal to 0.25 and 0.5. The horizontal line at zero delineates having exponentially many absorbing sets from the exponential absence of absorbing sets.

In Fig. 4.5 we plot the normalized logarithmic asymptotic elementary absorbing set distributions under the fixed ratio $\eta = \frac{\lambda}{\theta}$ for various $G_{n,m}^{\boldsymbol{3,r}}$. Although the ratio of unsatisfied checks and absorbing set size is fixed, as θ increases, the difference between trapping sets and absorbing sets increases. It follows that trapping sets can better approximate absorbing sets for smaller θ when $\eta = 0.5$. Following Fig. 4.5, Fig. 4.6 shows the normalized logarithmic asymptotic distributions of elementary trapping sets and absorbing sets in $G_{n,m}^{\boldsymbol{3,6}}$ under different ratios $\eta = \frac{\lambda}{\theta}$. Note that the curves corresponding to small values of η have the second zero crossing (see also Fig. 4.3). This second zero crossing can be used to represent the typical absorbing set distance (analogously to the typical trapping set distance, cf. [57]). It is also interesting to observe that the absorbing set curves taper off in Fig. 4.6. As η increases towards 1, it again becomes more difficult to meet the combinatorial constraints of an absorbing set.

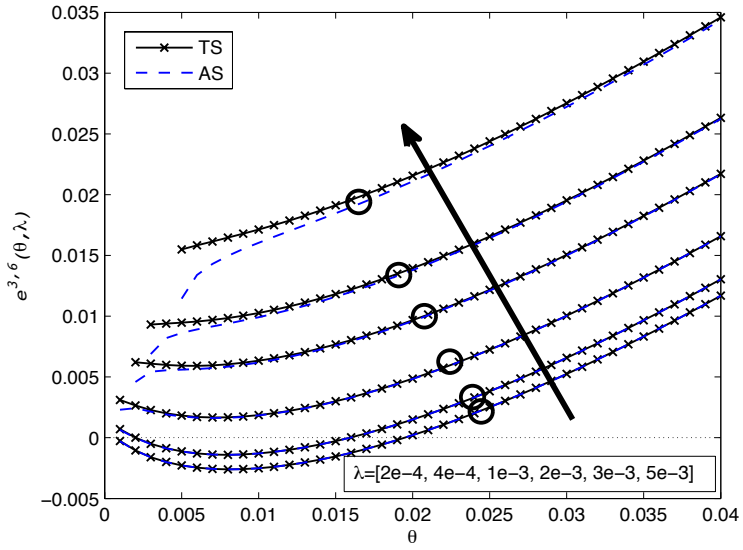


Figure 4.3: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in $G_{n,m}^{3,6}$ for different values of λ. The arrow indicates the increase in λ and the circles group up curves of the same λ. The equations in [17] are used to plot the trapping set curves.

Since the trapping set distribution is an upper bound of the absorbing set distribution, we also remark that certain known structured LDPC codes that have excellent minimum distance properties likely also have excellent absorbing set properties. In particular, repeat accumulate accumulate (RAA) codes of rates $1/3$ and below, have minimum distance growing linearly with blocklength [58, 59]. By comparing the results obtained here for the absorbing set analysis with the trapping set analysis of RAA codes presented in [60], we conclude that RAA of rates $1/3$ have substantially better absorbing set asymptotics than the random ensemble with bit node 3 and check node 9 for small values of θ and λ. See Figure 4.7.

LDPC code ensembles built out of protographs are another class of high-performance structured graph-based codes. Asymptotic enumeration of trapping sets was derived in [57] where it was shown that $(3, 6)$ protograph-based LDPC codes asymptotically behave the same as regular LDPC codes with the same bit and check node degree.

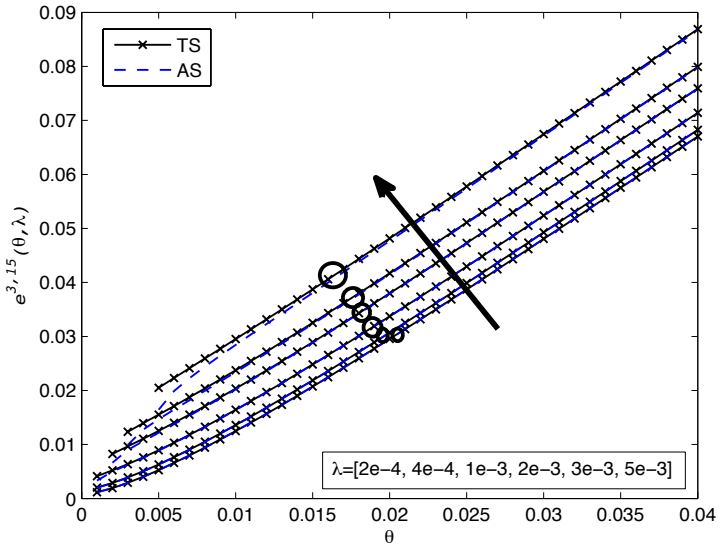


Figure 4.4: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in $G_{n,m}^{3,15}$ collection for different values of λ. The arrow indicates the increase in λ and the circles group up curves of the same λ.

Therefore these codes too should have asymptotic absorbing set properties no worse than the (unstructured) $(3, 6)$ regular ensemble.

4.2.3 Asymptotic Distribution of Non-Binary Absorbing Sets

Based on Theorem 1 in Chapter 3, we have the following corollary. Note that in order to distinguish unlabeled absorbing sets and non-binary absorbing sets, we refer to the weight assignments obeying the conditions in Definition 8 as *problematic*.

Corollary 1. *For regular code ensembles with column weight c and a given unlabeled (a, b) elementary absorbing set, the fraction of problematic weight assignments is given as $(q-1)^{\frac{b-(c-2)a-2}{2}}$.*

We now consider the asymptotic distribution of the non-binary absorbing sets. Let $\mathcal{G}_{n,m,q}^{c,r}$ denote the regular code ensemble with parity check matrices from the set $\Lambda_{n,m,q}^{c,r}$

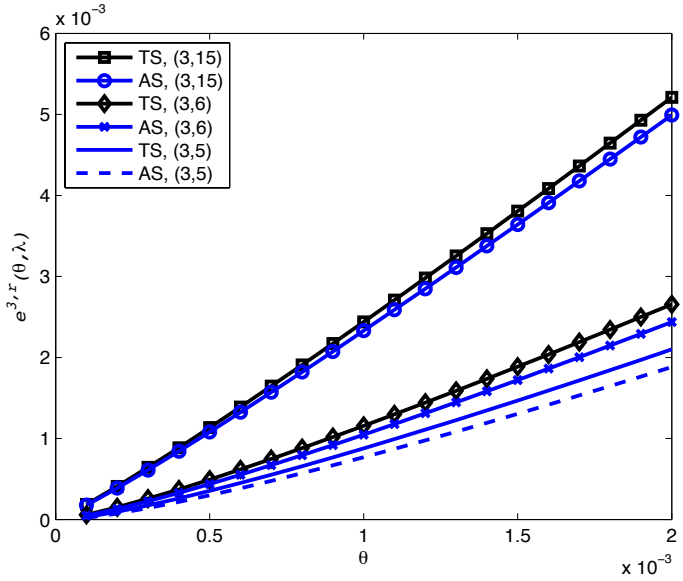


Figure 4.5: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in the Tanner graphs in $G^{3,r}_{n,m}$ for the fixed ratio $\frac{\lambda}{\theta} = 0.5$. The equations in [17] are used to plot the trapping set curves.

which consists of all regular matrices of column-weight c and row weight r with elements from GF(q). The asymptotic distribution of non-binary absorbing sets in the regular code ensemble is defined as follows:

$$e^{c,r}(\theta,\lambda) \triangleq \lim_{n\to\infty} \frac{1}{n} \log \frac{z^{c,r}_{a,b,n}}{|\Lambda^{c,r}_{n,m,q}|}, \tag{4.24}$$

where $\theta = \frac{a}{n}$, $\lambda = \frac{b}{n}$, and $z^{c,r}_{a,b,n}$ is the number of (a,b) absorbing sets over all matrices in $\Lambda^{c,r}_{n,m,q}$. In (4.24) and in the following, $\log$ is taken to the base e.

The following lemma specifies the normalized asymptotic distribution for the $c = 4$ case.

Lemma 3. *The normalized logarithmic asymptotic distribution of* $(a = \theta n, b = \lambda n)$ *elementary absorbing sets in* $\mathcal{G}^{4,r}_{n,m}$ *regular code ensemble over GF(q) is given as*

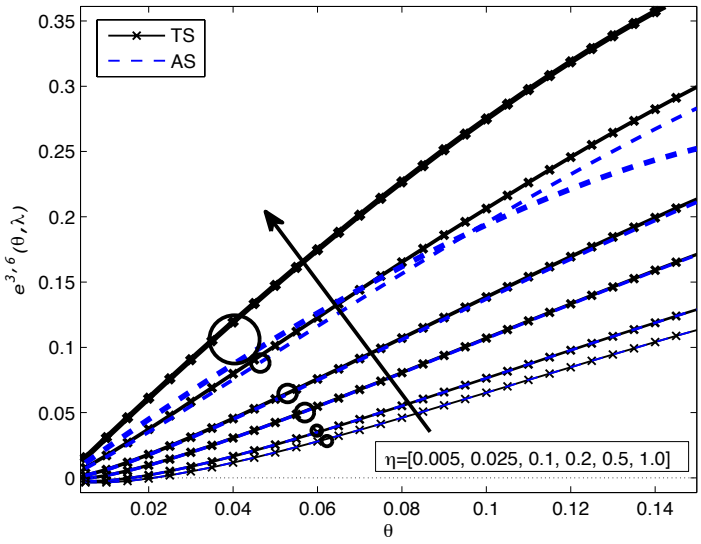


Figure 4.6: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in $G_{n,m}^{\boldsymbol{3,6}}$ for different $\eta = \frac{\lambda}{\theta}$. Thicker lines correspond to increasing values of η, as the arrow indicates. The circles group up curves of the same η. The equations in [17] are used to plot the trapping set curves.

$$e^{4,r}(\theta,\lambda) = -3H_b(\theta,1-\theta) - H_b(\zeta,1-\zeta) + \lambda\log 4r + \frac{4\theta-\lambda}{2}\log\binom{r}{2} + \theta H_b\left(\frac{\lambda}{\theta},1-\frac{\lambda}{\theta}\right) + H_b\left(1-\zeta,\lambda,\frac{4\theta-\lambda}{2},\frac{2\zeta-4\theta-\lambda}{2}\right) - 4\theta H_b\left(\frac{\lambda}{4\theta},1-\frac{\lambda}{4\theta}\right) - \frac{2\theta-\lambda}{2}log(q-1), \tag{4.25}$$

where $\zeta = \frac{m}{n}$*, and* $H_b(p_1,\cdots,p_N) = -\sum_{i=1}^{N} p_i \log p_i$ *with* $\sum_{i=1}^{N} p_i = 1$ *denotes the entropy function. The equation for* $e^{3,r}(\theta,\lambda)$ *can be found similarly.*

Proof. Using Corollary 1, from all unlabeled (binary) $(\theta n, \lambda n)$ absorbing sets in the Tanner graphs of regular code ensembles (enumerated in [29]), a fraction of $(q-1)^{\frac{(\lambda-(c-2)\theta)n-2}{2}}$ of all possible weight assignments results in $(\theta n, \lambda n)$ non-binary absorbing sets. For the details of the proof please see the Appendix. □

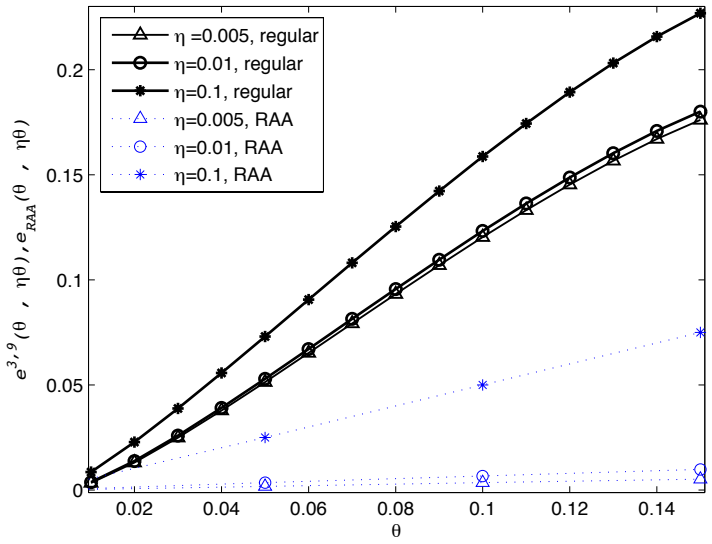


Figure 4.7: Comparison of the normalized logarithmic asymptotic distributions of elementary absorbing/trapping sets of rate $1/3$ regular unstructured codes ($e^{3,9}(\theta, \lambda)$) and of trapping sets of ($e_{RAA}(\theta, \eta\theta)$) rate $1/3$ RAA codes. Note the substantial improvement in the normalized trapping set distribution offered by the RAA codes relative to the lower bound of the regular LDPC code ensemble. The bound is based on the elementary absorbing set distribution.

Note that (4.25) indicates that the normalized asymptotic number of absorbing sets in the ensemble is a *decreasing* function in the field size q for fixed θ and λ.

For a $(3, 6)$ regular ensemble, Figure 4.8 shows the asymptotic distribution of unlabeled $(\theta n, \lambda n)$ elementary (binary) trapping sets using the results in [17], as well as $(\theta n, \lambda n)$ problematic elementary absorbing sets for $q = 2, 16$ and fixed $\theta = 0.001$. Note that all unlabeled $(\theta n, \lambda n)$ elementary (binary) trapping sets will result in $(\theta n, \lambda n^{+})$ non-binary trapping sets after labeling of the Tanner graph. We can observe that in particular for larger λ and for $q = 16$, the normalized asymptotic distribution of elementary absorbing sets is smaller than the one for elementary trapping sets. The reason for this behavior is that after the weight assignment in the non-binary case a smaller fraction of unlabeled trapping sets leads to problematic absorbing sets. Therefore, the

resulting non-binary absorbing set enumerators provide a better assessment of the error floor compared to trapping set enumerators. Also, as the value of λ gets smaller, both curves for unlabeled elementary trapping sets and binary elementary absorbing set converge, which means that most of the unlabeled elementary trapping sets result in problematic elementary absorbing sets in this case.

Figure 4.9(a) shows the asymptotic distribution of absorbing sets for regular $(3, 6)$ ensemble and fixed $\lambda/\theta \approx 1$. In this case, the distributions are similar for different values of q, since the absorbing sets include only one (fundamental) cycle and as a a result, there is one constraint on the edge weights of absorbing sets. Therefore, the difference between the asymptotic distributions curves for different values of q is negligible in logarithmic domain. On the other hand, in Figure 4.9(b) where $\lambda/\theta = 0.1$, the number of satisfied checks and consequently the number of fundamental cycles in the VN graph increases, which puts more constraints on edge weights to be problematic. This leads to an increasing gap between the curves for the binary and the non-binary cases. We also observe that an increase of q results in a smaller number of normalized asymptotic absorbing sets. The intuition for this behavior is that as q increases, the number of possible weight assignments also increases for an unlabeled structure and as a result, a smaller fraction of weight assignments satisfy the condition in Lemma 1. Figure 4.9(b) also demonstrates that as q increases, the second zero crossing of the absorbing set growth rate curve, i.e., the typical relative smallest absorbing set, increases as well.

4.3 Analysis and Enumeration of Non-Binary Protograph-Based LDPC Codes

In this section, we generalize existing definitions and techniques from the binary to the non-binary domain and provide ensemble performance evaluation of the resultant non-binary protograph-based (NBPB) codes through the explicit computation of codeword

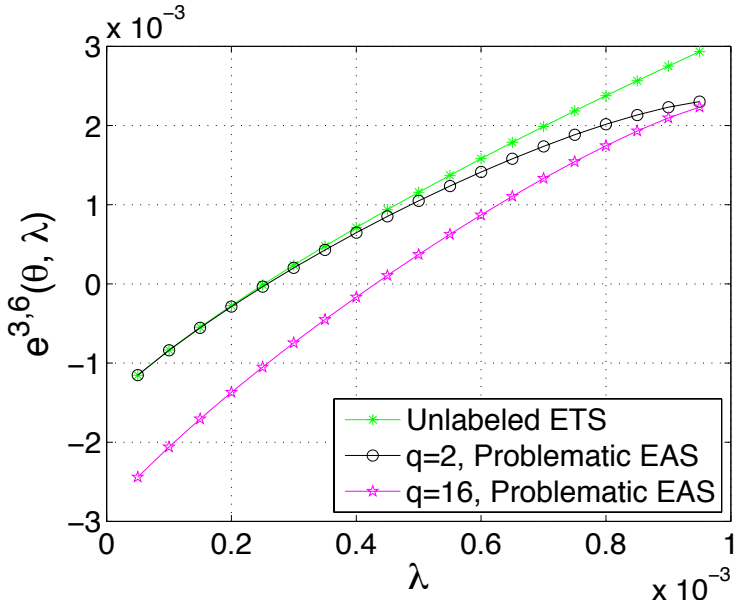
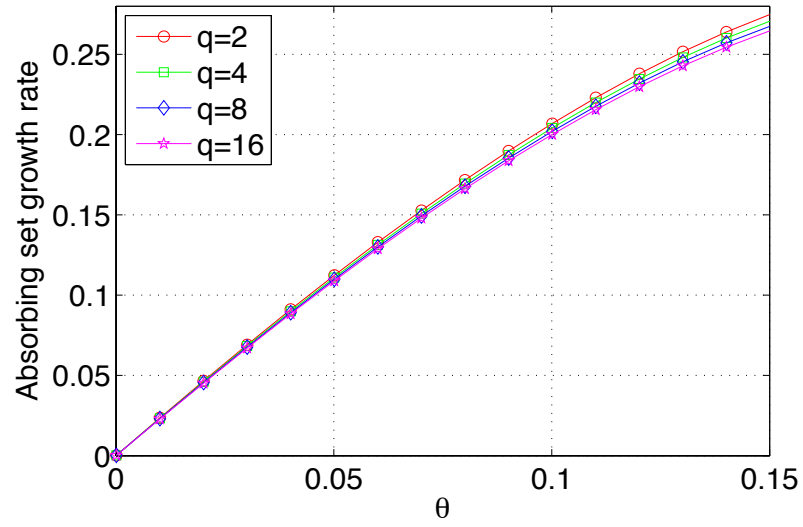


Figure 4.8: Normalized logarithmic asymptotic distributions of elementary trapping sets (ETS) and absorbing sets (EAS) for $G_{n,m}^{3,6}$, $\theta = 0.001$ and $q = 2, 16$.

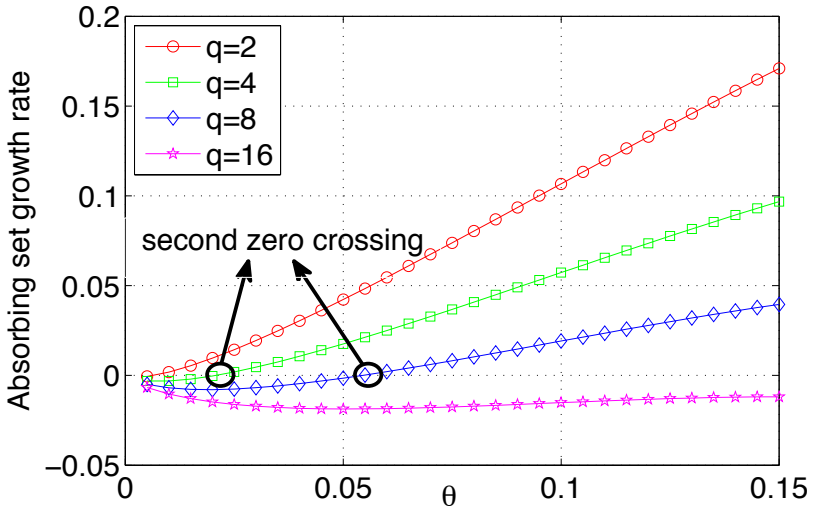
enumerator.

4.3.1 U-NBPB Weight Enumerators

In this subsection, we compute the codeword weight enumerators which are known to be useful for bounding the performance under the maximum likelihood (ML) decoding. The section is composed of three parts. We first provide the exact weight enumerator of a code induced by one check node (Subsection 4.3.1.1). We then discuss non-asymptotic ensemble weight enumerators (Subsection 4.3.1.2) and the asymptotic ensemble weight enumerators (Subsection 4.3.1.3). Some of the presented results build upon [33], and generalize these known results to the non-binary set-up. Throughout the section, illustrative examples accompany the derivations.



(a) $\lambda/\theta = 0.95$



(b) $\lambda/\theta = 0.1$

Figure 4.9: Normalized logarithmic asymptotic distributions of elementary absorbing sets for regular code ensemble $G^{3,6}_{n,m}$, $q = 2, 4, 8, 16$ and (a) $\lambda/\theta = 0.95$, (b) $\lambda/\theta = 0.1$.

4.3.1.1 Weight Enumerator of A Check Node and Its Replicas

Let us begin building the enumerator result by first considering a check node c_j with degree m_j in the mother protograph G. We first establish the necessary notation.

It is convenient to view this check node c_j as a $(m_j,\, m_j - 1)$ linear block code $\mathcal{C}_j$ over $GF(q)$. Let $K_j = q^{(m_j-1)}$ denote the number of codewords in $\mathcal{C}_j$. Further, let $\mathbf{M}^{\mathcal{C}_j}$ be the $K_j \times m_j$ matrix with the codewords of $\mathcal{C}_j$ as its rows (whose entries are by construction in $GF(q)$), and let $\mathbf{M}_b^{\mathcal{C}_j}$ be the $K_j \times m_j$ *binary matrix* obtained by converting all non-zero entries of $\mathbf{M}^{\mathcal{C}_j}$ to 1. Note that by construction, some rows of $\mathbf{M}_b^{\mathcal{C}_j}$ may be the same. Let the collection $\mathcal{M}_b^{\mathcal{C}_j}$ represent all rows $\mathbf{x}$ of $\mathbf{M}_b^{\mathcal{C}_j}$, where $\mathbf{x} = [x_1, x_2, \ldots, x_{m_j}]$, $x_i \in \{0, 1\}$. Define a $K_{j,r} \times m_j$ *binary matrix* $\mathbf{M}_{b,r}^{\mathcal{C}_j}$ as the submatrix of $\mathbf{M}_b^{\mathcal{C}_j}$ that consists of all distinct rows of $\mathbf{M}_b^{\mathcal{C}_j}$. The number of rows in $\mathbf{M}_{b,r}^{\mathcal{C}_j}$ is $K_{j,r} = 1 + \sum_{i=2}^{m_j} \binom{m_j}{i}$. Let the set $\mathcal{M}_{b,r}^{\mathcal{C}_j}$ represent the rows $\mathbf{x}_k = [x_{k,1}, x_{k,2}, \ldots, x_{k,m_j}]$, $x_{k,i} \in \{0, 1\}$, for $i = 1, 2, \ldots m_j, k = 1, 2, \ldots, K_{j,r}$ of $\mathbf{M}_{b,r}^{\mathcal{C}_j}$.

Following the proposed construction of U-NBPB codes, we consider the N copies of node c_j in the daughter graph, and call the resultant $(Nm_j,\, N(m_j - 1))$ linear block code $\mathcal{C}_j^N$. It is convenient to denote by n_k the number of occurrences of the k^{th} codeword among these N copies of c_j, and to collect them into the vector $\mathbf{n}$, where $\mathbf{n} = [n_1, n_2, \ldots, n_{K_j}]$. Lastly, let $A^{\mathcal{C}_j^N}(\mathbf{w})$ denote the weight-vector enumerator of $\mathcal{C}_j^N$ where $\mathbf{w} = [w_1, w_2, \ldots, w_{m_j}]$ is the weight vector of the input message in $\mathcal{C}_j^N$, where the entry w_i denotes the number of occurrences of a non-zero value in position i, $1 \leq i \leq m_j$, over the set of input messages.

With the above, the main result of this subsection is provided in the following theorem that characterizes the weight enumerator of the code $\mathcal{C}_j^N$ (in the daughter graph G^N) that is described by N copies of the single check node c_j (in the mother graph G). For the ease of exposition and since we currently focus on the single check node, we suppress the index j in c_j, $\mathcal{C}_j^N$ and $K_{j,r}$, and simply refer to the check node as c, its resultant code as $\mathcal{C}^N$, and reduced row dimension as K_r.

Throughout the analysis

$$C\left(N; x_1, x_2, \cdots, x_L\right) = \frac{N!}{x_1! x_2! \cdots x_L!}, \tag{4.26}$$

denotes the multinomial coefficient, where x_i's are non-negative integers summing to

N.

Theorem 5. *The weight-vector enumerator $A^{\mathcal{C}^N}(\mathbf{w})$ of $\mathcal{C}^N$ is given by,*

$$A^{\mathcal{C}^N}(\mathbf{w}) = \sum_{\{\mathbf{n}\}} C\left(N; n_1, n_2, \ldots, n_{K_r}\right) e^{\mathbf{n}\cdot\mathbf{f}_q^T}, \tag{4.27}$$

where $C\left(N; n_1, n_2, \ldots, n_{K_r}\right)$ is the multinomial coefficient specified in (4.26)*, and $\{\mathbf{n}\}$ is the set of integer-vector solutions to $\mathbf{w} = \mathbf{n}\cdot\mathbf{M}_{b,r}^{\mathcal{C}}$, with $n_1, n_2, \ldots, n_{K_r} \geq 0$, and $\sum_{k=1}^{K_r} n_k = N$. The vector $\mathbf{f}_q = [f_{q,1}, f_{q,2} \ldots, f_{q,K_r}]$ has entries $f_{q,k} = \ln g(q, |\mathbf{x}_k|)$, where $\mathbf{x}_k$ is the k-th element of $\mathcal{M}_{b,r}^{\mathcal{C}}$, $|\mathbf{x}_k|$ is the weight of $\mathbf{x}_k$, and $g(q,i) = \frac{q-1}{q}[(q-1)^{i-1} + (-1)^i]$.*

Proof. The weight-vector enumerators $\{A^{\mathcal{C}^N}(\mathbf{w})\}$ may be found as the coefficients of a multi-dimensional generating function of $\{A^{\mathcal{C}^N}(\mathbf{w})\}$. The generating function of the code $\mathcal{C}$ induced by the check node c is $\sum_{\mathbf{x}\in\mathcal{M}_b^{\mathcal{C}}} W_1^{x_1} W_2^{x_2} \cdots W_m^{x_m}$, where the W_i's are indeterminate bookkeeping variables.

From [32], the weight generating function for the code $\mathcal{C}$ induced by a single check node c of degree m, is given by $A^{\mathcal{C}}(W) = \frac{1}{q}[(1+(q-1)W)^m + (q-1)(1-W)^m]$, which also holds for $GF(q)$. This generating function can also be written as $A^{\mathcal{C}}(W) = \sum_{w=0}^{m} \binom{m}{w} g(q,w) W^w$. For our problem, the number $g(q,w)$ represents exactly the number of repeated rows with weight w in $\mathcal{M}_b^{\mathcal{C}}$. Thus, $\sum_{\mathbf{x}\in\mathcal{M}_b^{\mathcal{C}}} W_1^{x_1} W_2^{x_2} \cdots W_m^{x_m} = \sum_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} g(q, |\mathbf{x}_k|) W_1^{x_{k,1}} W_2^{x_{k,2}} \cdots W_m^{x_{k,m}}$, where $\mathbf{x}_k$ is the k-th element of $\mathcal{M}_{b,r}^{\mathcal{C}}$ and $|\mathbf{x}_k|$ is its weight (that is, the sum of its entries). The generating function for N copies of this check node in the daughter graph is then

$$A^{\mathcal{C}^N}(W_1, W_2, \ldots, W_m) = \left(\sum_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} g(q, |\mathbf{x}_k|)\, W_1^{x_{k,1}} W_2^{x_{k,2}} \cdots W_m^{x_{k,m}} \right)^N . \tag{4.28}$$

Applying the multinomial theorem to (4.28) yields,

$$A^{\mathcal{C}^N}(W_1, W_2, \ldots, W_m) = \sum_{\substack{n_1,n_2,\ldots,n_{K_r}\geq 0 \\ n_1+n_2+\cdots+n_{K_r}=N}} C\left(N; n_1, n_2, \ldots, n_{K_r}\right) \times \prod_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} \left(g(q, |\mathbf{x}_k|) W_1^{x_{k,1}} W_2^{x_{k,2}} \cdots W_m^{x_{k,m}}\right)^{n_k}. \tag{4.29}$$

Then, (4.29) can be written as

$$A^{\mathcal{C}^N}(W_1, W_2, \ldots, W_m) = \sum_{\mathbf{w}} \sum_{\{\mathbf{n}\}} C\left(N; n_1, n_2, \ldots, n_{K_r}\right) \times \left(\prod_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} \left[g(q, |\mathbf{x}_k|)\right]^{n_k} \right) \times W_1^{w_1} W_2^{w_2} \cdots W_m^{w_m}, \tag{4.30}$$

where $\{\mathbf{n}\}$ is the set of integer solutions to $\mathbf{w} = \mathbf{n} \cdot \mathbf{M}_{b,r}^{\mathcal{C}}$, under the constraints $n_1, n_2, \ldots, n_{K_r} \geq 0$ and $\sum_{k=1}^{K_r} n_k = N$, and where $w_l = \sum_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} x_{k,l} n_k$, $l = \{1, 2, \ldots, m\}$. To see the last step, note that the product in (4.29) can be manipulated as follows

$$\prod_{\forall \mathbf{x}_k \in \mathcal{M}_{b,r}^{\mathcal{C}}} \left(W_1^{x_{k,1}} W_2^{x_{k,2}} \cdots W_m^{x_{k,m}}\right)^{n_k} = W_1^{w_1} W_2^{w_2} \cdots W_m^{w_m}. \tag{4.31}$$

Also, if $\mathbf{w} = \mathbf{n} \cdot \mathbf{M}_{b,r}^{\mathcal{C}}$ has more than one solution for $\mathbf{n}$, the term $W_1^{w_1} W_2^{w_2} \cdots W_m^{w_m}$ will appear as a common factor in all of the terms that are associated with these solutions. This observation explains the presence of the second summation in (4.30). The generating function of $\{A^{\mathcal{C}^N}(\mathbf{w})\}$ can also be written as

$$A^{\mathcal{C}^N}(W_1, W_2, \ldots, W_m) = \sum_{\mathbf{w}} A^{\mathcal{C}^N}(\mathbf{w}) W_1^{w_1} W_2^{w_2} \cdots W_m^{w_m}. \tag{4.32}$$

Finally, comparing (4.32) and (4.30) leads to (4.27). □

Note that if we choose to use $\mathcal{M}_b^{\mathcal{C}}$ (which has repeated elements) then

$$A^{\mathcal{C}^N}(\mathbf{w}) = \sum_{\{\mathbf{n}\}} C\left(N; n_1, n_2, \ldots, n_K\right), \tag{4.33}$$

where $\{\mathbf{n}\}$ is now the set of integer-vector solutions to $\mathbf{w} = \mathbf{n} \cdot \mathbf{M}_b^{\mathcal{C}}$, with $n_1, n_2, \ldots, n_K \geq 0$, $\sum_{k=1}^{K} n_k = N$, and $K = q^{m-1}$. We now provide an illustrative example.

Example 3. *Consider a $(3, 2)$ linear block code over $GF(q)$ replicated N times, whose weight enumerator we seek to compute. There is only one check node so we simply refer to this node as c and to the code it generates as $\mathcal{C}$. Let $\mathcal{C}^N$ denote the $(3N, 2N)$ code obtained by replicating $\mathcal{C}$ code N times. Our objective is to evaluate $A^{\mathcal{C}^N}(w_1, w_2, w_3)$.*

We now show that if we start with (4.33) we can in fact obtain (4.27) with reduced computational complexity. Observe that $M_b^{\mathcal{C}}$ is a $q^2 \times 3$ (binary) matrix with repeated rows. Solving the equation $\mathbf{w} = \mathbf{n} \cdot \mathbf{M}_b^{\mathcal{C}}$ for $K = q^2$ integers n_i, $i \in \{1, 2, \ldots, K\}$, only requires to solve for 5 integers.

In the set of codewords (x_1, x_2, x_3) of this $(3, 2)$ code, apart from the all-zeros codeword, there are $(q-1)$ codewords of Hamming weight 2, where x_i and x_j are non-zero, and x_k is zero element of $GF(q)$, for i, j, k distinct indices from the set $\{1, 2, 3\}$. There are also $(q-1)(q-2)$ codewords of Hamming weight 3. The set $\mathcal{M}_{b,r}^{\mathcal{C}}$ is $\{[0, 0, 0], [0, 1, 1], [1, 0, 1], [1, 1, 0], [1, 1, 1]\}$ and the matrix $\mathbf{M}_{b,r}^{\mathcal{C}}$ (the reduced version of the matrix $\mathbf{M}_b^{\mathcal{C}}$) is then the lexicographical ordering of these rows.

Computing the solution to $\mathbf{w} = \mathbf{n} \cdot \mathbf{M}_b^{\mathcal{C}}$ is equivalent to solving the set of equations $\mathbf{w} = \mathbf{k} \cdot \mathbf{M}_{b,r}^{\mathcal{C}}$, $n_1 = k_1$, $\sum_{i=2}^{q} n_i = k_2$, $\sum_{i=q+1}^{2q-1} n_i = k_3$, $\sum_{i=2q}^{3q-2} n_i = k_4$, $\sum_{i=3q-1}^{q^2} n_i = k_5$, where $\mathbf{k} = [k_1, k_2, k_3, k_4, k_5]$, and $\mathbf{w} = [w_1, w_2, w_3]$. An application of the multinomial theorem results in $\sum_{\substack{i_1, i_2, \ldots, i_l \geq 0 \\ i_1 + i_2 + \cdots + i_l = t}} \frac{1}{i_1! i_2! \ldots i_l!} = \frac{l^t}{t!}$. Using this equality, one can show that (4.33) reduces to

$$A^{\mathcal{C}^N}(\mathbf{w}) = \sum_{\{\mathbf{k}\}} C\left(N; k_1, \ldots, k_5\right) (q-1)^{k_2 + k_3 + k_4 + k_5} (q-2)^{k_5}, \tag{4.34}$$

where $\{\mathbf{k}\}$ is the set of integer-vector solutions to $\mathbf{w} = \mathbf{k} \cdot \mathbf{M}_{b,r}^{\mathcal{C}}$, with $k_1, k_2, \ldots, k_5 \geq 0$ and $\sum_{i=1}^{5} k_i = N$. Solving this set of equations we get $k_1 = N - s + k_5/2$, $k_2 = s - w_1 - k_5/2$, $k_3 = s - w_2 - k_5/2$, and $k_4 = s - w_3 - k_5/2$, where $s = (w_1 + w_2 + w_3)/2$. Since $k_i \geq 0$, we have $\max\{0, 2(s-N)\} \leq k_5 \leq 2s - 2\max\{w_1, w_2, w_3\}$.

If $w_1 + w_2 + w_3$ is even, then

$$\begin{aligned} A^{\mathcal{C}^N}(\mathbf{w}) = \textstyle\sum_l C\left(N; (N-s+l), (s-w_1-l),\right. \\ \left.(s-w_2-l), (s-w_3-l), (2l)\right) \times (q-1)^{(s-l)}(q-2)^{2l}, \end{aligned} \tag{4.35}$$

where $l = k_5/2$ and k_5 is even. If $w_1 + w_2 + w_3$ is odd, then

$$\begin{aligned} A^{\mathcal{C}^N}(w) = \textstyle\sum_l C\big(N; (N-s+l+1/2), (s-w_1-l-1/2),\\ (s-w_2-l-1/2), (s-w_3-l-1/2), (2l+1)\big)\\ \times (q-1)^{(s-l-1/2)}(q-2)^{2l+1}, \end{aligned} \tag{4.36}$$

where $l = (k_5-1)/2$ and k_5 is odd. ■

Based on this exact combinatorial count on the per-node basis, in the next section we derive the exact weight enumerator of the U-NBPB ensemble.

4.3.1.2 Weight Enumerator of the U-NBPB Ensemble

Before stating the enumerator result, we first define the non-binary uniform interleaver.

Definition 9 (Non-binary uniform interleaver)**.** *A length-L non-binary uniform interleaver over $GF(q)$ is a probabilistic device that maps each input of length L and of Hamming weight w into the $(q-1)^w\binom{L}{w}$ distinct weighted permutations of the input, such that it generates each weighted permutation with equal probability, $\frac{1}{(q-1)^w\binom{L}{w}}$.* ■

The notion of Uniform Codeword Selector (UCS) was introduced in [61] in the context of the concatenation of non-binary product codes. This definition is equivalent to the notion of non-binary uniform interleaver in this paper.

With the protograph based set-up, it is convenient to view the resultant code as a serial concatenation of certain component codes (*cf.* [62]). Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ are two serially concatenated block codes over $GF(q)$ that are connected by a length-L non-binary uniform interleaver over $GF(q)$. For the given codes $\mathcal{C}_1$ and $\mathcal{C}_2$, let $SCC = SCC(\mathcal{C}_1, \mathcal{C}_2)$ be the resultant ensemble over all possible interleavers.

Lemma 4. *Consider two block codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of dimensions K_l and codeword lengths N_l, $l = 1, 2$, that are serially concatenated via a non-binary uniform interleaver, with all system components over $GF(q)$. The average number of codewords of Hamming weight d that are created by inputs of Hamming weight f in the resultant SCC ensemble is given by*

$$A_{f,d}^{SCC} = \sum_{w} \frac{A_{f,w}^{\mathcal{C}_1} A_{w,d}^{\mathcal{C}_2}}{(q-1)^w \binom{K_2}{w}}, \tag{4.37}$$

where $A_{f,w}^{\mathcal{C}_1}$ is the number of codewords in $\mathcal{C}_1$ of Hamming weight w corresponding to $\mathcal{C}_1$-encoder inputs of Hamming weight f, and $A_{w,d}^{\mathcal{C}_2}$ is the number of codewords in $\mathcal{C}_2$ of Hamming weight d corresponding to C_2-encoder inputs of Hamming weight w.

Proof. For the constituent code $\mathcal{C}_1$, there are $(q-1)^f \binom{K_1}{f}$ possible encoder inputs of Hamming weight f, and they produce $A_{f,w}^{\mathcal{C}_1}$ codewords of Hamming weight w. Likewise, for the constituent code $\mathcal{C}_2$, there are $(q-1)^w \binom{K_2}{w}$ possible encoder inputs of weight w, and they produce $A_{w,d}^{\mathcal{C}_2}$ codewords of Hamming weight d. When $\mathcal{C}_1$ and $\mathcal{C}_2$ are connected via a length-K_2 non-binary uniform interleaver ($K_2 = N_1$), each of these $A_{f,w}^{\mathcal{C}_1}$ codewords in $\mathcal{C}_1$ maps into one of the $A_{w,d}^{\mathcal{C}_2}$ codewords of $\mathcal{C}_2$ with probability $\frac{1}{(q-1)^w \binom{K_2}{w}}$.

Averaged over the resultant SCC ensemble, there are $A_{f,w}^{\mathcal{C}_1} A_{w,d}^{\mathcal{C}_2} / (q-1)^w \binom{K_2}{w}$ codewords of Hamming weight d corresponding to the SCC encoder inputs of weight f and to the $\mathcal{C}_2$-encoder inputs of weight w. Summing these codewords over all w, (4.37) follows. □

Based on Lemma 4 we derive the exact weight enumerator over the entire U-NBPB ensemble as follows.

Recall that there are n_v variable nodes and n_c check nodes in the mother protograph G, and that m_j denotes the degree of the check node c_j. Let t_i denote the degree of the variable node v_i. Recall that the U-NBPB ensemble consists of all codes obtained by performing all possible weight permutations on the edges of the daughter graph G^N.

Let $\mathbf{d}_j = \left[d_{j_1}, d_{j_2}, ..., d_{j_{m_j}}\right]$ be the weight vector which describes the weights of the N symbol words on the edges connected to check node c_j, produced by the variable nodes $\{v_{j_1}, v_{j_2}, ..., v_{j_{m_j}}\}$ neighboring c_j.

It is convenient to specify Kronecker Delta $\kappa_{x,y}$ as

$$\kappa_{x,y} = \begin{cases} 1 & \text{if } x = y, \text{ and} \\ 0 & \text{if} x \neq y. \end{cases} \tag{4.38}$$

If x and y are vectors, we interpret Kronecker Delta having value 1 only if x and y agree in all components.

Theorem 6. *The weight-vector enumerator of the U-NBPB code averaged over the entire ensemble is*

$$A(\mathbf{d}) \;=\; \frac{\prod_{j=1}^{n_c} A^{\mathcal{C}_j^N}(\mathbf{d}_j)}{\prod_{i=1}^{n_v} (q-1)^{d_i(t_i-1)} \binom{N}{d_i}^{t_i-1}}, \tag{4.39}$$

where $A^{\mathcal{C}_j^N}(\mathbf{d}_j)$ is the weight-vector enumerator of the code $\mathcal{C}_j^N$ induced by the N copies of the check node c_j. Here, the elements of $\mathbf{d}_j$ comprise a subset of the elements of $\mathbf{d} = [d_1, d_2, ..., d_{n_v}]$, and this subset is obtained from the edge connections in the mother protograph G (see Fig. 4.10 for illustration).

Proof. Consider N copies of each node in the protograph as a constituent code. These constituent codes are then inter-connected through non-binary uniform interleavers each of size $N \times N$. The N copies of each variable node $v_i \in G$ can be treated as a constituent code with one input of weight d_i and t_i outputs $[w_{i,1}, w_{i,2}, \ldots, w_{i,t_i}]$. The input-output weight coefficient for node v_i is then $(q-1)^{d_i}\binom{N}{d_i}\kappa_{d_i,w_{i,1}} \cdots \kappa_{d_i,w_{i,t_i}}$. The N copies of each check node $c_j \in G$ can be treated as a constituent code with m_j input weights $\mathbf{w_j} = [w_{j,1}, w_{j,2}, \ldots, w_{j,m_j}]$ and no output.

Let $A^{\mathcal{C}_j^N}(\mathbf{w}_j)$ be the input weight enumerator of the check node group $\mathcal{C}_j^N$. Let $A(\mathbf{d})$ represent the number of sequences each with weight vector $\mathbf{d} = [d_1, d_2, \ldots, d_{n_v}]$ that is applied to the variable nodes according to the protograph constraints.

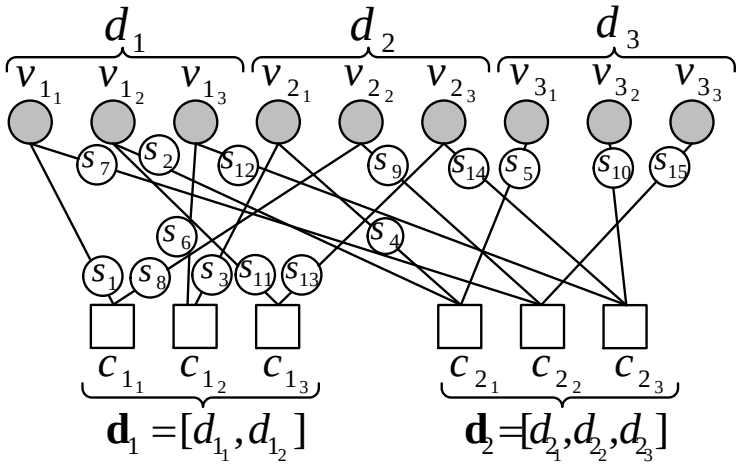


Figure 4.10: Illustration of the relationship between vectors $\mathbf{d} = [d_1, d_2, d_3]$ and $\mathbf{d_1} = [d_{1_1}, d_{1_2}]$, $\mathbf{d_2} = [d_{2_1}, d_{2_2}, d_{2_3}]$ for an U-NBPB code with $N = 3$, where $d_{1_1} = d_1$, $d_{1_2} = d_2$, $d_{2_1} = d_1$, $d_{2_2} = d_2$, $d_{2_3} = d_3$.

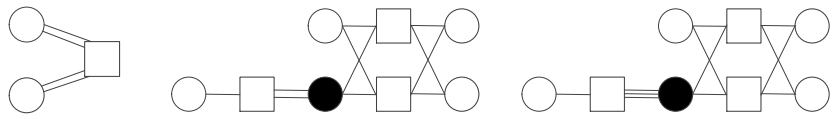
Then, the result of Lemma 4 is applied to individual concatenations to obtain the average protograph weight-vector enumerator as,

$$\begin{aligned} A(\mathbf{d}) &= \sum_{\substack{w_{m,u} \\ m=1,\ldots,n_v \\ u=1,\ldots,t_m}} \frac{\prod_{k=1}^{n_v} [(q-1)^{d_k} \binom{N}{d_k} g_{d_k,w_{k,1}} \cdots g_{d_k,w_{k,t_k}}]}{\prod_{s=1}^{n_v} \prod_{r=1}^{t_s} (q-1)^{w_{s,r}} \binom{N}{w_{s,r}}} \\ &\times \prod_{i=1}^{n_c} A^{\mathcal{C}_j^N}(\mathbf{w}_j). \end{aligned} \tag{4.40}$$

Here, the summation is over all weights $w_{m,u}$, where $w_{m,u}$ is the weight along the u^{th} edge of variable node v_m. Note that $w_{j,l} = w_{i,k}$ if the l^{th} edge of check node c_j is the k^{th} edge of variable node v_i. The vector $\mathbf{d}_j = \left[d_{j_1}, d_{j_2}, ..., d_{j_{m_j}}\right]$ is a weight vector which describes the weights of the N-symbol words on the edges connected to check node c_j, produced by the variable nodes neighboring c_j. The elements of $\mathbf{d}_j$ comprise a subset of the elements of $\mathbf{d}$. Then, (4.40) reduces to

$$A(\mathbf{d}) = \frac{\prod_{j=1}^{n_c} A^{\mathcal{C}_j^N}(\mathbf{d}_j)}{\prod_{i=1}^{n_v} (q-1)^{d_i(t_i-1)} \binom{N}{d_i}^{t_i-1}},$$

as desired. □



(a) (b) (c)

Figure 4.11: Three candidate protographs: (a) Regular $(2, 4)$ protograph, (b) Punctured $(2, 4)$ type 1 protograph, and (c) Punctured $(2, 4)$ type 2 protograph. Black nodes are punctured.

The average number of codewords of symbol weight d in the ensemble, denoted by A_d, equals the sum of $A(\mathbf{d})$ over all $\mathbf{d}$ for which $\sum_{\{d_i:v_i\in V\}} d_i = d$.

Example 4. *In this example we calculate the symbol weight enumerator for the three protographs given in Fig. 4.11. The first protograph describes a regular* $(2, 4)$ *code, the second and the third protographs are obtained by adding an accumulator to the regular* $(2, 4)$ *protograph followed by puncturing of a node. We refer to the former as the punctured* $(2, 4)$ *type 1 protograph and we refer to the latter as the punctured* $(2, 4)$ *type 2 protograph. Here and in subsequent examples black nodes are punctured. In the calculations, all three code ensembles have* 32 *transmitted variable nodes and are defined over* $GF(8)$*, so the total number of bits is* 96*. As a result, the first code has* $N = 16$*, and the second and the third code have* $N = 8$*. The results for the average weight enumerator* A_d *are shown in Fig. 4.12 for the smallest* 9 *non-zero codeword symbol weights. To further illustrate the enumeration technique, we plot the weight enumerators of the three protograph code ensembles with* 80 *transmitted variable nodes over* $GF(8)$*, i.e.,* $N = 40$ *for the first code and* $N = 20$ *for the second and third codes. The results are shown in Fig. 4.13, also for the lowest* 9 *non-zero codeword symbol weights. We note that, relative to the regular* $(2, 4)$ *code, the punctured type 1 code and the punctured type 2 code both have on average fewer low symbol weight codewords, and that the type 2 code has the best distribution of the three codes for small codeword weights.*

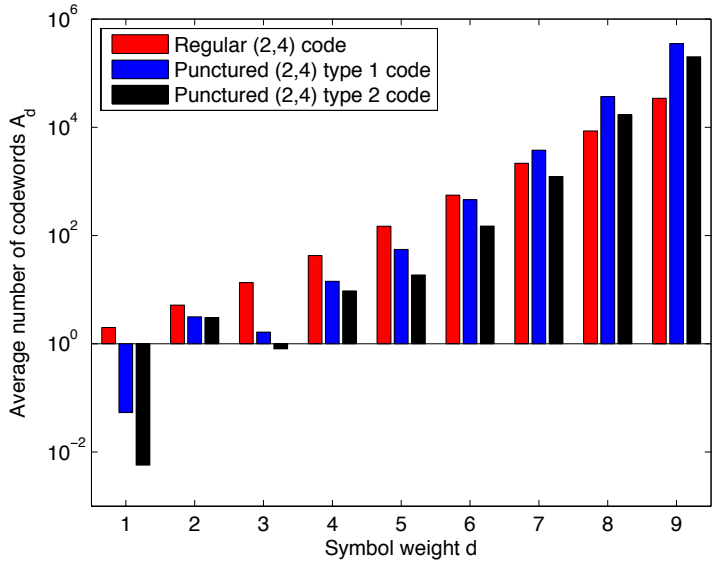


Figure 4.12: Weight enumerator for the U-NBPB ensembles of the protographs in Fig. 4.11 over $GF(8)$ for symbol length 32.

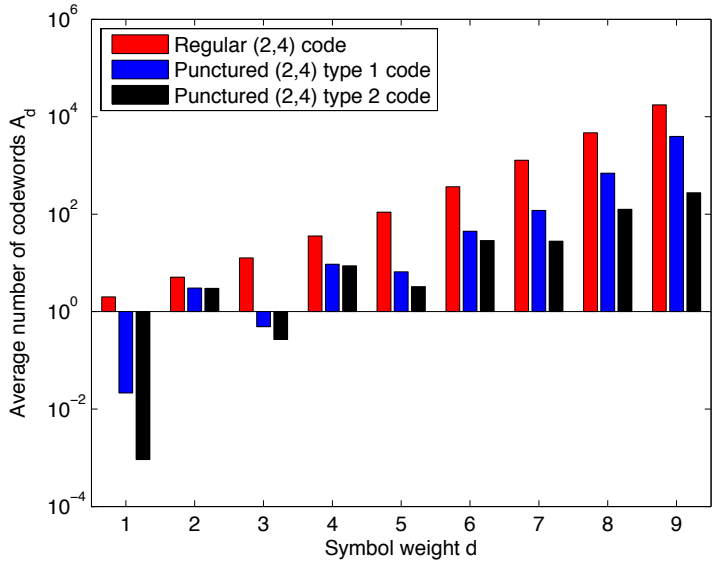


Figure 4.13: Weight enumerator for the U-NBPB ensembles of the protographs in Fig. 4.11 over $GF(8)$ for symbol length 80.

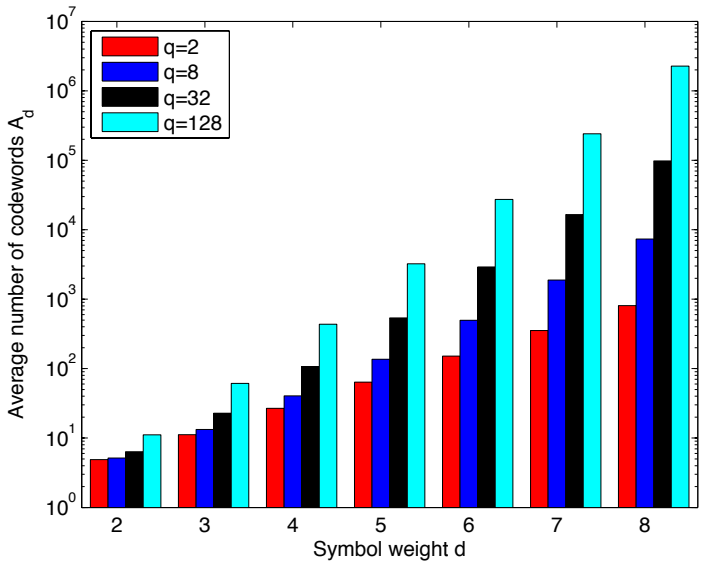


Figure 4.14: Weight enumerator for the U-NBPB ensembles of the regular $(2, 4)$ protograph in Fig. 4.11 for symbol length 40 and over different field orders.

Example 5. *Continuing on with the baseline regular* $(2, 4)$ *protograph repeated* $N = 20$ *times (i.e., with* 40 *symbols), in Fig. 4.14, we now plot the average number of codewords,* A_d*, as a function of the field order* q *for the first few smallest values of the non-zero symbol weight. As expected, the average number of codewords increases with* q*.*

4.3.1.3 Asymptotic Ensemble Weight Enumerators

Given that the formulas in the previous subsection involve the number of copies N, we define the normalized logarithmic asymptotic weight (the growth rate) to be

$$r(\delta) = \limsup_{N\to\infty} \frac{\ln A_d}{N} = \limsup_{N\to\infty} \frac{\ln A_{\delta N}}{N}, \tag{4.41}$$

where $\delta = d/N$. Note that $n = n_v \cdot N$, so the growth rate in terms of n can be expressed as

$$\tilde{r}(\tilde{\delta}) = \limsup_{n\to\infty} \frac{\ln A_d}{n}, \tag{4.42}$$

where $\tilde{r}(\tilde{\delta}) = \frac{1}{n_v} r(\tilde{\delta} n_v)$.

From (4.39), we have

$$\ln A(\mathbf{d}) = \sum_{j=1}^{n_c} \ln A^{\mathcal{C}_j^N}(\mathbf{d}_j) - \sum_{i=1}^{n_v} (t_i - 1) \left[d_i \ln(q-1) + \ln \binom{N}{d_i} \right]. \tag{4.43}$$

Let $\delta_i = d_i/N$, and take the limit as $N \to \infty$. Using $\limsup_{N\to\infty} \ln \binom{N}{d_i}/N = H(\delta_i) = -(1-\delta_i)\ln(1-\delta_i) - \delta_i \ln \delta_i$, [31], we obtain

$$r(\delta) = \max_{\{\delta_l : v_l \in V\}} \left\{ \sum_{j=1}^{n_c} a^{c_j}(\delta_j) - \sum_{i=1}^{n_v} (t_i - 1)[H_q(\delta_i)] \right\}, \tag{4.44}$$

under the constraint $\sum_{\{\delta_i : v_i \in V\}} \delta_i = \delta$, and $H_q(\delta_i) \triangleq \delta_i \ln(q-1) + H(\delta_i)$. In (4.44), $a^{c_j}(\delta_j)$ is the asymptotic weight-vector enumerator associated with the check node c_j, defined as

$$a^{c_j}(\omega) = \limsup_{N\to\infty} \frac{\ln A^{\mathcal{C}_j^N}(\mathbf{w})}{N}, \tag{4.45}$$

where $\omega = \mathbf{w}/N$, and $\delta_j = \mathbf{d}_j/N$.

Let $P_{\boldsymbol{\omega}} = [p_1, p_2, \ldots, p_K]$ be the relative proportion of occurrences of each codeword of constituent check node code $\mathcal{C}$ in a sequence of N codewords, where $p_k = n_k/N$ and n_k is the number of occurrences of the k^{th} codeword. We then let the type class of $P_{\boldsymbol{\omega}}$, $T(P_{\boldsymbol{\omega}})$, be the set of all length-N sequences of codewords in $\mathcal{C}$, each containing n_k occurrences of the k^{th} codeword in $\mathcal{C}$, for $k = 1, 2, ..., K_r$. Observe that $|T(P_{\boldsymbol{\omega}})| = C(N; n_1, n_2, \ldots, n_{K_r})$. From [31, Thm.12.1.3] and [33], $|T(P_{\boldsymbol{\omega}})| \to e^{N \cdot H(P_{\boldsymbol{\omega}})}$, as $N \to \infty$, where $H(P_{\boldsymbol{\omega}}) = -\sum_{k=1}^{K_r} p_k \ln p_k$. As $N \to \infty$ (4.27) is

$$\begin{aligned} A^{\mathcal{C}}(\mathbf{w}) &= \textstyle\sum_{\{\mathbf{n}\}} C(N; n_1, n_2, \ldots, n_{K_r})\, e^{\mathbf{n}\cdot\mathbf{f}_q^T} \\ &= \textstyle\sum_{\{P_{\boldsymbol{\omega}}\}} |T(P_{\boldsymbol{\omega}})| e^{N P_{\boldsymbol{\omega}}\cdot\mathbf{f}_q^T} \to \sum_{\{P_{\boldsymbol{\omega}}\}} e^{N[H(P_{\boldsymbol{\omega}}) + P_{\boldsymbol{\omega}}\cdot\mathbf{f}_q^T]}, \end{aligned}$$

under the constraint that $\{P_{\boldsymbol{\omega}}\}$ is the set of solutions to $\boldsymbol{\omega} = P_{\boldsymbol{\omega}} \cdot \mathbf{M}_{b,r}^{\mathcal{C}}$, with $p_1, p_2, \ldots, p_{K_r} \geq 0$ and $\sum_{k=1}^{K_r} p_k = 1$. It follows from (4.45) that

$$a^{\mathcal{C}}(\boldsymbol{\omega}) = \max_{\{P_{\boldsymbol{\omega}}\}} \left\{ H(P_{\boldsymbol{\omega}}) + P_{\boldsymbol{\omega}} \cdot \mathbf{f}_q^T \right\}. \tag{4.46}$$

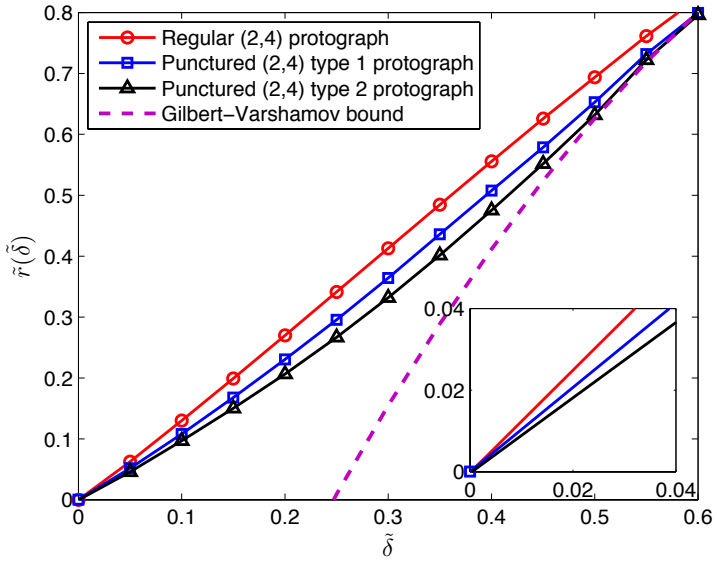


Figure 4.15: Asymptotic symbol weight enumerators of protographs in Fig. 4.11 over $GF(8)$.

Example 6. *Continuing with the protographs discussed in Example 4, we compute the asymptotic symbol weight enumerators for the three protographs for $q = 8$, as shown in Fig 4.15. As we can see, in the asymptotic case, the punctured type 1 protograph and the punctured type 2 protograph both have on average fewer low symbol weight codewords than the regular $(2, 4)$ protograph. This result is in agreement with the finite length calculation (and will be later shown to be also consistent with the threshold calculations).*

Note that the ensemble of all rate-R, q-ary ("random") linear codes (whose parity-check matrix entries are i.i.d. uniform) has the weight enumerator $A^{\mathcal{C}}(w) = (q-1)^w \binom{n}{w} e^{-n(1-R)\ln(q)}$ and the asymptotic weight enumerator [32]

$$\tilde{r}(\tilde{\delta}) = H_q(\tilde{\delta}) - (1-R)\ln(q), \tag{4.47}$$

which corresponds to the asymptotic Gilbert-Varshamov bound for the non-binary case. In Fig. 4.15, we plot the Gilbert-Varshamov bound for $q = 8$. Similar to the binary protograph case studied in [33], the asymptotic symbol weight enumerators converge to

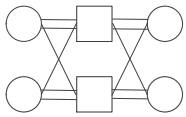


Figure 4.16: Regular $(3, 6)$ protograph.

the Gilbert-Varshamov bound as $\tilde{\delta}$ gets larger. Here, again, of the three candidate protographs, the punctured type 2 protograph offers the growth rate closest to the Gilbert-Varshamov bound.

Example 7. *In this example, we provide the asymptotic weight enumerator for the regular* $(3, 6)$ *protograph (presented in Fig. 4.16) over* $GF(q)$*, as shown in Fig. 4.17. We also note that our result for* $GF(2)$ *is in agreement with [33]. From the figure, we can see that as* q *increases, there are fewer low weight codewords. In addition, as* q *increases, the growth rate of high weight codewords increases. We use* ν_{min} *to denote the second zero crossing of* $\tilde{r}(\tilde{\delta})$ *(the first zero crossing is* $\tilde{r}(0) = 0$*). The second zero crossing, if it exists, is called the typical relative minimum distance.*

Fig. 4.18 shows how the typical relative minimum distance ν_{min} *changes with varying* q*. Consistent with [34], while the Gilbert -Varshamov bound grows monotonically with* q*,* ν_{min} *is in fact non-monotonic. In particular,* ν_{min} *attains maximum value for* $q = 64, 128$*.*

4.3.2 Trapping Set Enumerators for U-NBPB Ensembles

In this section we consider the trapping set enumerators of the U-NBPB.Let us consider a $T_{a,b}$ trapping set in the Tanner graph corresponding to the U-NBPB code (G, N, S_N, Π) over GF(q). In the spirit of the approach in [33], first we set the values of these a variable nodes in $T_{a,b}$ to (arbitrary) non-zero elements of GF(q) and set the values of all remaining variable nodes to the zero element of GF(q), so that b neighboring check nodes are unsatisfied. We then attach additional b variable nodes, one to each of these b

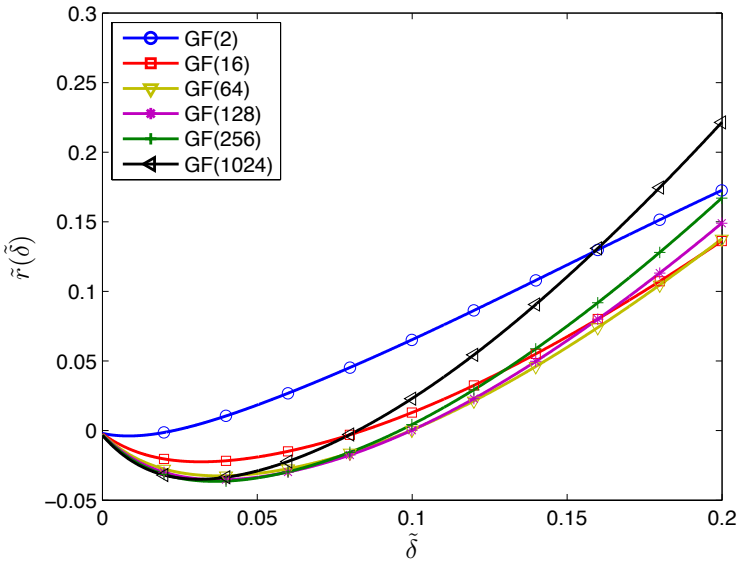


Figure 4.17: Asymptotic symbol weight enumerators of regular $(3, 6)$ protograph for different q.

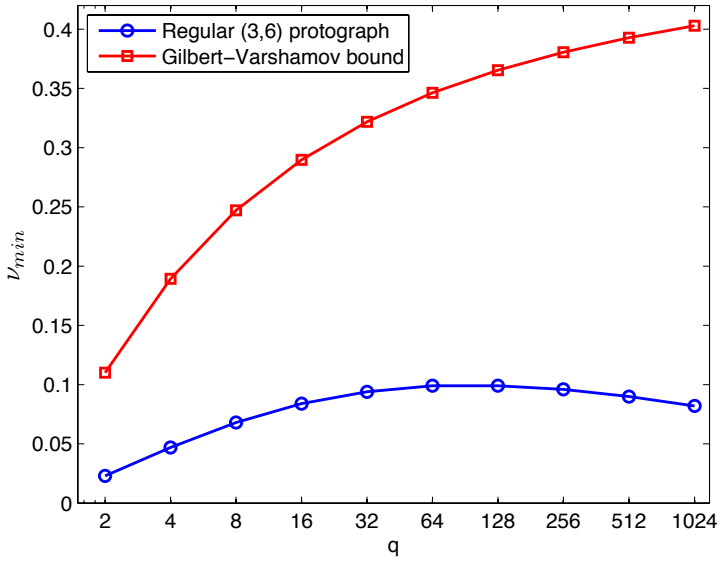


Figure 4.18: Typical minimum distance of regular $(3, 6)$ protograph for different q.

check nodes in the graph $\tilde{G}$. The attached nodes are connected via new edges of weight 1 each, and have a non-zero value uniquely chosen to force all check nodes to be satisfied. This operation can then be interpreted as suitably adding degree-1 variable nodes to all check nodes in the underlying protograph G. Let this set of nodes be F and call the new graph G'. We can then obtain the trapping set enumerator of the U-NBPB ensemble specified by G from the weight enumerator of the U-NBPB ensemble specified by G'. In particular, the $T_{a,b}$ trapping set enumerator $A_{a,b}^{(t)}$ is computed as

$$A_{a,b}^{(t)} = \sum_{\{d_i:v_i\in V\}} \sum_{\{d_k:v_k\in F\}} A(\mathbf{d}), \tag{4.48}$$

under the constraints $\sum_{\{d_i:v_i\in V\}} d_i = a$ and $\sum_{\{d_i:v_i\in F\}} d_i = b$, where

$$A(\mathbf{d}) = \frac{\prod_{j=1}^{n_c} A^{\mathcal{C}'^N_j}(\mathbf{d}_j)}{\prod_{i=1}^{n_v} (q-1)^{d_i(t_i-1)} \binom{N}{d_i}^{t_i-1}} \ . \tag{4.49}$$

We use $\mathcal{C}'^N_j$ instead of $\mathcal{C}_j^N$ in (4.49) to indicate that the weight vector enumerators in (4.49) are obtained from the check nodes in G'. These weight vector enumerators can be evaluated using (4.27).

As in Section 4.3.1.3, we define the normalized logarithmic asymptotic trapping set enumerator $\tilde{r}^{(t)}(\tilde{\alpha}, \tilde{\beta})$, as

$$\tilde{r}^{(t)}(\tilde{\alpha}, \tilde{\beta}) = \limsup_{n\to\infty} \frac{\ln A_{\tilde{\alpha}n,\tilde{\beta}n}^{(t)}}{n}, \tag{4.50}$$

for given $\tilde{\alpha} > 0$ and $\tilde{\beta} > 0$ (which are independent of n). The derivation of an expression for (4.50) from (4.49) uses the same steps used in deriving $\tilde{r}(\tilde{\delta})$, and yields

$$\tilde{r}^{(t)}(\tilde{\alpha}, \tilde{\beta}) = \frac{1}{n_v} r^{(t)}(\tilde{\alpha}n_v, \tilde{\beta}n_v), \tag{4.51}$$

where

$$r^{(t)}(\alpha, \beta) = \max_{\{\delta_l:v_l\in V\}} \Big\{ \max_{\{\delta_k:v_k\in F\}} \{ \sum_{j=1}^{n_c} a^{c'_j}(\delta_j) - \sum_{i=1}^{n_v} H_q(\delta_i) \} \Big\}, \tag{4.52}$$

under the constraints $\sum_{\{\delta_i:v_i\in V\}} \delta_i = \alpha$, and $\sum_{\{\delta_i:v_i\in F\}} \delta_i = \beta$. The asymptotic weight vector enumerator, $a^{c'_j}(\delta_j)$, can be evaluated using (4.46).

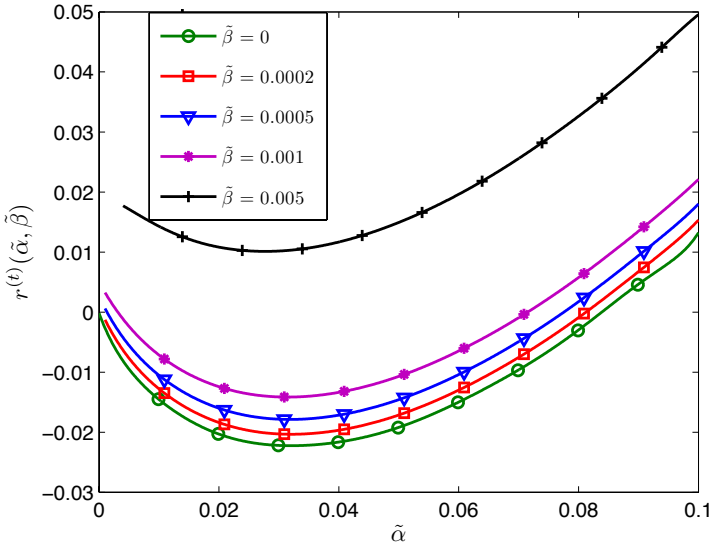


Figure 4.19: Asymptotic trapping set enumerators of the $(3, 6)$-regular protograph code ensemble over GF(16).

Example 8. *Let us consider the* $(3, 6)$*-regular protograph code ensemble over GF(*16*). The asymptotic trapping set enumerators are plotted for different* $\tilde{\beta}$ *in Fig. 4.19. Note when* $\tilde{\beta} = 0$*, by our definition, the curve corresponds to the asymptotic symbol weight enumerator of the* $(3, 6)$*-regular protograph. In the figure, when* $\tilde{\alpha}$ *is fixed,* $\tilde{r}^{(t)}(\tilde{\alpha}, \tilde{\beta})$ *increases with increasing* $\tilde{\beta}$*. This result is consistent with the trapping set enumerator for binary protograph-based LDPC codes reported in [33].*

Example 9. *In this example, we consider the* $(3, 6)$*-regular protograph code ensemble for different* q*'s with fixed* $\tilde{\beta} = 0.0002$*. The asymptotic enumeration results are shown in Fig. 4.20. In the figure, we can see that when* $\tilde{\beta} = 0.0002$*, there always exists the second zero-crossing, i.e., there exist the typical relative* $\tilde{r}^{(t)}(\tilde{\alpha}, 0.0002)$ *smallest trapping sets for different* q*'s. Also, when* $\tilde{\alpha}$ *is fixed,* $\tilde{r}^{(t)}(\tilde{\alpha}, 0.0002)$ *decreases as* q *increases. This indicates that for* $\tilde{\beta} = 0.0002$ *(and more generally), codes over larger* q *have fewer trapping sets.*

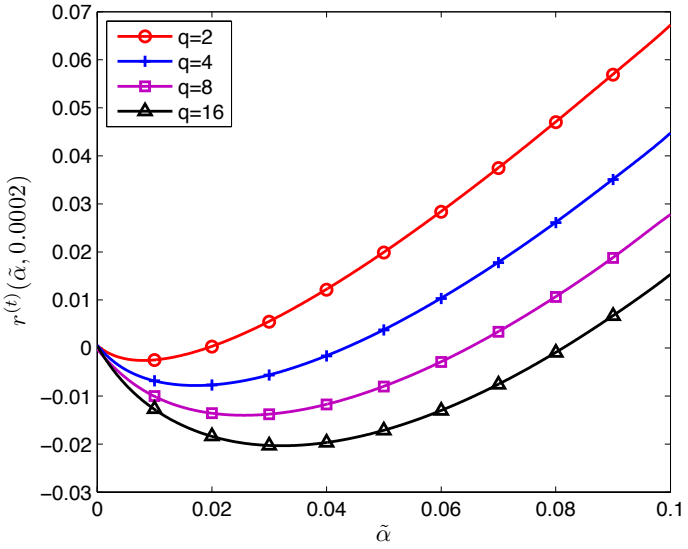


Figure 4.20: Asymptotic trapping set enumerators of the $(3, 6)$-regular protograph code ensemble for different q with $\tilde{\beta} = 0.0002$, for protograph shown in Fig. 4.16.

CHAPTER 5

Finite-Length Analysis and Design of Spatially-Coupled Codes

In Chapters 3 and 4, we studied finite-length and asymptotic properties of non-binary LDPC block codes. In this chapter, we consider a new class of graph-based codes, called spatially-coupled codes, which have shown to have desirable properties for data storage applications. In the asymptotic regime, spatially-coupled codes are known to reach capacity-approaching threshold under belief-propagation decoding, which implies outstanding finite-length performance required in storage applications. Furthermore, the introduction of lower complexity sliding window decoding makes spatially-couple codes well-suited for practical applications.

This section provides mathematical analysis for finite-length error floor performance of spatially-coupled codes. By employing our analysis, we offer spatially-coupled codes which have outstanding performance in low-error rate regions targeted in data storage applications. Our study is a first step to design spatially-coupled code systems applicable to future data storage systems.

5.1 Introduction

In this chapter, and inspired in part by [63], we present a complete characterization of absorbing sets for binary and non-binary array-based spatially-coupled (AB-SC) codes. Our contributions include:

1. We introduce an analytical approach to find the exact number of absorbing sets in AB-SC codes: the original counting problem is mapped to a problem of finding the number of integer points within an area in 2D space.

2. We find the optimal cutting vector for AB-SC codes with arbitrary circulant size: analytical and experimental results reveal that the choice of the cutting vector significantly affects the error floor performance of binary AB-SC codes.

3. We calculate the average number of absorbing sets in non-binary AB-SC codes constructed by uninformed (random) assignment of edge weights on top of a binary AB-SC code. This result reveals that the average number of absorbing sets in non-binary AB-SC codes is significantly lower than the average number of absorbing sets in binary AB-SC codes. This explains the superior error floor performance of non-binary AB-SC codes compared to their binary counterparts.

5.2 Construction of Array-Based Spatially-Coupled LDPC Codes

In this section, we review the construction of AB-SC codes [63]. Let p be a prime number indicating the circulant size and row weight, c be the column weight, and L be the coupling length. Moreover, let $\boldsymbol{\xi} = [\xi_0, \cdots, \xi_{c-1}]$ be the cutting vector [1] where $0 \leq \xi_0 < \xi_1 < \cdots < \xi_{c-1} \leq p$. We first construct the $cp \times p^2$ underlying array-based LDPC (AB-LDPC) code as follows:

$$H(c,p) = \begin{bmatrix} I & I & I & ... & I \\ I & \sigma & \sigma^2 & \cdots & \sigma^{(p-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & \sigma^{(c-1)} & \sigma^{2(c-1)} & \cdots & \sigma^{(p-1)(c-1)} \end{bmatrix}, \tag{5.1}$$

where σ is a $p \times p$ circulant matrix formed by cyclically shifting all rows of the identity matrix one element to the left. Remember that the matrix $H(c,p)$ can be viewed as a

[1] We assume that $\xi_i, i \in \{0, 1, \cdots, c-1\}$ are distinct. In the general case, these parameters are not necessarily distinct.

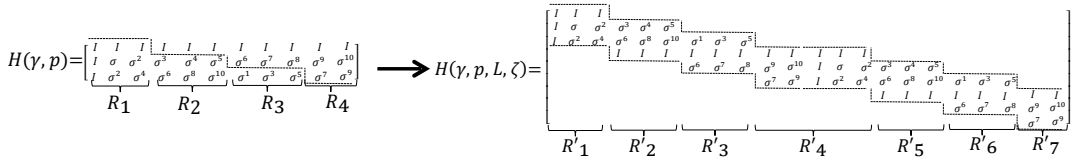
2-D array of submatrices where each row (column) of matrices denotes a row (column) group $i, 0 \leq i \leq c-1$ ($j, 0 \leq j \leq p-1$). For our discussions, we describe each column of $H(c,p)$ by a pair (j,k) where j is the index of the column group, and k, $0 \leq k \leq p-1$, is the index of the column within the column group.

Based on the given cutting vector $\boldsymbol{\xi}$, the matrix H_0 of size $cp \times p^2$ is formed by assigning each circulant matrix in row group i and column group j, $j < \xi_i$, of $H(c,p)$ to the equivalent position in H_0. All other remaining elements of H_0 are then set to 0. Furthermore, the matrix H_1 is defined as $H_1 = H(c,p) - H_0$.

The parity-check matrix of an AB-SC code with given coupling length L is then defined as

$$H(c,p,L,\boldsymbol{\xi}) \triangleq \begin{bmatrix} H_0 & 0 & 0 & \cdots & 0 & 0 \\ H_1 & H_0 & 0 & \cdots & 0 & 0 \\ 0 & H_1 & H_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_1 & H_0 \\ 0 & 0 & 0 & \cdots & 0 & H_1 \end{bmatrix}. \tag{5.2}$$

For the considered construction, the constraint length, which is the maximal width of the non-zero area in each row of $H(c,p,L,\boldsymbol{\xi})$, is equal to $\nu_s = p^2$. Note that in the above construction the syndrome former memory m_s is assumed to be 1. In the general case, where $m_s \geq 1$, we have $H(c,p) = H_0 + H_1 + \cdots + H_{m_s}$. Figure 5.1 shows an example of an AB-SC construction where $c = 3$, $p = 11$, $L = 2$ and $\boldsymbol{\xi} = [3,6,9]$. For the sake of our later discussion, we define the region $R_n, n \in \{1, \cdots, c+1\}$ in $H(c,p)$ as the set of column groups with indices between ξ_{n-2} and ξ_{n-1} (cmp. Figure 5.1(a)). We assume $\xi_{-1} = 0$ and $\xi_c = p$, and the number of column groups within the region $R_n, n \in \{1, \cdots, c+1\}$ is denoted by r_n. The definition of regions can be expanded to AB-SC codes, where each region is similarly defined as the set of column groups between two consecutive edges of the cutting vector (cmp. Figure 5.1(b)).



(a) (b)

Figure 5.1: (a) Example of an AB-LDPC code with $p = 11$ and $c = 3$, (b) Example of an AB-SC code with $p = 11$, $c = 3$, $L = 2$, and cutting vector $\boldsymbol{\xi} = [3, 6, 9]$. Here, $r_i = |R_i|$ for $i \in \{1, 2, 3, 4\}$ and $r'_i = |R'_i|$ for $i \in \{1, 2, \cdots, 7\}$. Moreover, $r'_1 = r_1$, $r'_2 = r_2$, $r'_3 = r_3$, $r'_4 = r_4 + r_1$, $r'_5 = r_2$, $r'_6 = r_3$, and $r'_7 = r_4$.

The following remark describes the construction of non-binary AB-SC codes.

Remark 2. *For a given matrix $H(c, p)$ of a binary AB-LDPC code, the parity-check matrix $H_q(c, p)$ of a non-binary AB-LDPC code over GF(q) can be constructed by replacing the elements with value '1' of $H(c, p)$ with non-zero elements of GF(q). The parity-check matrix $H_q(c, p, L, \boldsymbol{\xi})$ of a non-binary AB-SC code is then constructed by the same edge spreading procedure as above.*

5.3 Performance Comparison for AB-LDPC and AB-SC Codes

In the following, we study the error floor performance of spatially-coupled codes. We unveil the properties of the error profile spectrum of these codes and exploit these properties to optimize the code design. In particular, we will show the following properties: a) AB-SC codes have better performance in the low error rate (error floor) region compared to AB-LDPC codes. This observation motivates us to characterize absorbing sets in AB-SC codes and present an exact enumeration of small absorbing sets in these codes. Based on these results we then show that due to partitioning of the underlying parity check matrix into H_0 and H_1 components, some absorbing sets are broken and thus do not exist in the Tanner graph of the resulting AB-SC code. The reduction in

the number of small problematic absorbing sets in moving from block to SC designs can in part explain the superior error floor performance of AB-SC codes compared to AB-LDPC codes.

b) The choice of the cutting vector significantly affects the performance of binary AB-SC codes in the low error rate region. Through our analysis we find the exact number of small absorbing sets as a function of the cutting vector. This analysis enables us to optimize the choice of the cutting vector to design AB-SC codes with the minimum number of problematic absorbing sets.

c) The error profile spectra are different for AB-LDPC and AB-SC codes. We show that the number of dominant absorbing sets in the error profile of an array-based code significantly change as we apply the coupling process.

Before we lay down the theoretical framework for this analysis we motivate the work with the following example.

Example 10. *Figure* 2 *shows simulation results*[2] *for an AB-LDPC code with block length* 4489 *bits, circulant size* 67*, column weight* 3*, and rate* ≈ 0.9*. We also present the performance curves for comparable AB-SC codes with same constraint length* $\nu_s = 4489$*, and column weight* 3*. Note that distinct cutting vectors* [3] *are chosen uniformly at random for the construction of each AB-SC code shown in Figure* 2*. Table 5.1 includes the error profiles, i.e., the number of specific absorbing sets, for the curves in Figure* 2 *at an SNR of* 6.1*dB. Figure* 2 *shows that AB-SC codes have a performance improvement of around one order of magnitude compared to AB-LDPC codes. We also observe that different cutting vectors result in a change of the error floor performance of AB-SC codes. This observation suggests that the number of problematic absorbing sets in the Tanner graph of SB-SC codes changes with the choice of the cutting vector.*

[2] *All the simulation results presented in this paper were performed on the Hoffman2 Cluster which is a part of High Performance Computing Resources at UCLA. The Hoffman2 cluster has more than* 800 *machines and about* 7000 *cores. The CPUs have* 8*,* 12 *or* 16 *cores with speed of* $2.2 - 3.0$ *GHz. Each core has* 1*GB,* 4*GB or* 8*GB of memory.*

[3] *For code 1,* $\boldsymbol{\xi} = [10, 18, 56]$*, for code 2,* $\boldsymbol{\xi} = [22, 28, 55]$*, for code 3,* $\boldsymbol{\xi} = [8, 31, 40]$*, and for code 4,* $\boldsymbol{\xi} = [15, 31, 47]$.

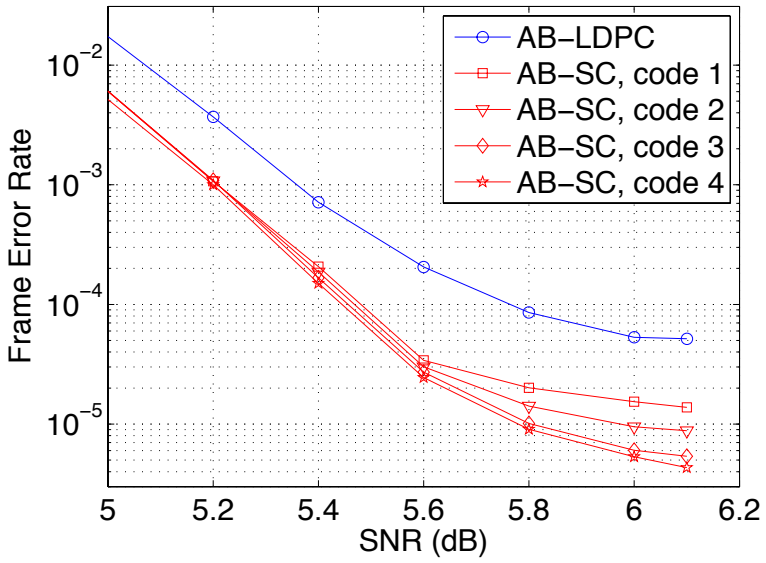


Figure 5.2: Performance comparison for AB-LDPC and AB-SC codes with various random cutting vectors.

Table 5.1: Error profile (the number of specific absorbing sets) for the curves shown in Figure 2 at an SNR of 6.1 dB.

Error Type	$(3,3)$	$(4,2)$	$(4,4)$	$(5,1)$	$(5,3)$	$(5,5)$	other
AB-LDPC	55	23	13	2	3	8	17
AB-SC, code 1	71	6	8	0	4	3	21

Table 5.1 also shows that the distribution of absorbing set errors is different for AB-LDPC and AB-SC codes. For exampe, the percentage of $(4,4)$ *absorbing set errors are significantly different (19% vs. 5.3%). Our forthcoming analysis will mathematically justify the above observations.*

5.4 Exact Enumeration of Binary Absorbing Sets in Array-Based Spatially-Coupled Codes

In this section, we first observe that the structure of AB-SC codes imposes additional constraints (relative to the block case) on two variable nodes sharing a check in an absorbing set. We then introduce an approach to calculate the exact number of absorbing sets for binary AB-SC codes. Although our procedure is applicable to any column weight, for the ease of discussion we limit our analysis to column weight 3 and 4 codes.

We first revisit the bit, check, and pattern consistency conditions for AB-LDPC codes from [16].

Lemma 4. ***Bit consistency:*** *The neighboring CNs of a VN must have distinct row-group indices i.*

Check consistency: *The neighboring VNs of a CN must have distinct column-group indices j.*

Pattern consistency: *If two VNs corresponding to columns (j_1, k_1) and (j_2, k_2) share a CN c_1 in row group i_1, then*

$$k_1 + i_1 j_1 \equiv k_2 + i_1 j_2 \mod p. \tag{5.3}$$

Remark 3. *In a spatially-coupled structure, each CN can only be connected to VNs within a window of at most c consecutive regions[4], where c is the column weight of the code. As an example, each CN in the parity-check matrix shown in Figure 5.1(b) is connected to the VNs in a window of at most $c = 3$ regions. In addition to the pattern consistency condition, this property imposes extra constraints on the valid choices for column groups of two VNs sharing a CN.*

The following example clarifies Remark 3.

Example 11. *Consider the $H(3, 11, 2, [3, 6, 9])$ parity-check matrix in Figure 5.1(b). If two VNs share a CN in the second row group ($i = 1$), then the regions which the*

[4] *Note that the width of the window is shorter for the first and last $c - 1$ columns in $H(c, p, L, \boldsymbol{\xi})$.*

two VNs belong to must be within the set $\{R'_1, R'_2\}$. *Assuming that* v_1 *belongs to* R'_1 *with width* $r_1 = 3$ *and that* v_2 *belongs to* R'_2 *with width* $r_2 = 3$*, the following three conditions must be satisfied:*

$$k_1 + j_1 = k_2 + j_2 \mod p, \quad 0 \leq j_1 \leq r_1 - 1, \quad r_1 \leq j_2 \leq r_1 + r_2 - 1.$$

5.4.1 Column Weight 3 Analysis

In this subsection, we present the exact number of minimal absorbing sets in $c = 3$ AB-SC codes. The following lemma presents the size of minimal absorbing sets in AB-SC codes with column weight 3.

Lemma 5. ([63]) *The smallest possible absorbing sets in the Tanner graph of* $H(3, p, L, \boldsymbol{\xi})$ *are* $(3, 3)$ *and* $(4, 2)$ *absorbing sets.*

Since small absorbing sets typically contribute the most to the error floors of LDPC codes, we focus our analysis on $(3, 3)$ and $(4, 2)$ absorbing sets. In [63], the authors show that the number of minimal absorbing sets in column weight 3 AB-SC codes grows linearly with the coupling length L. Through our analysis, we provide the exact number of $(3, 3)$ and $(4, 2)$ absorbing sets in the Tanner graph of $H(3, p, L, \boldsymbol{\xi})$ as a function of circulant size p, coupling length L, and the cutting vector $\boldsymbol{\xi}$.

5.4.1.1 Analysis of $(3, 3)$ absorbing sets

We consider the $(3, 3)$ absorbing set structure shown in Figure 5.3. Without loss of generality, we assume that v_1 and v_2 share a check in the row group $i_1 = 0$, v_2 and v_3 share a check in the row group $i_2 = 1$, and v_1 and v_3 share a check in the row group $i_3 = 2$. Thus, the pattern consistency constraints lead to:

$$k_1 + 2j_1 \equiv k_3 + 2j_3, \quad k_1 = k_2, \quad k_2 + j_2 \equiv k_3 + j_3, \tag{5.4}$$

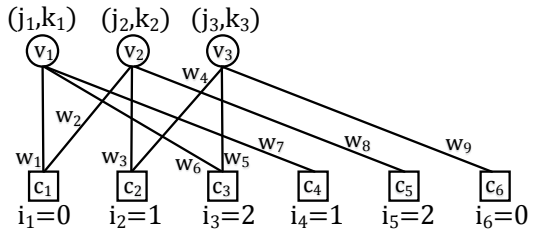


Figure 5.3: Structure of a $(3, 3)$ absorbing sets over GF(q). Note that in the binary case, w_1 through w_9 are equal to '1'.

where j_1, j_2, j_3, k_1, k_2, and k_3 are in $\{0, 1, \cdots, p-1\}$ and all equations are modulo p. The above equations results in the following equation, involving only the indices of the column groups:

$$j_2 \equiv 2j_1 - j_3 \mod p. \tag{5.5}$$

It was shown in [16] that by fixing the values of j_1, j_3, and k_1 in the above equations, the values of all other variables can be uniquely determined. In the case of AB-LDPC codes, j_1 and j_3 can take any pair of distinct values between 0 and $p-1$. Index k_1 also can take any integer value between 0 and $p-1$. Therefore there exist $p^2(p-1)$ absorbing sets of size $(3, 3)$ in column weight 3 AB-LDPC codes. Through our analysis below, we show that not all pairs of (j_1, j_3) are valid in the case of AB-SC codes; this constraint results in a fewer number of $(3, 3)$ absorbing sets. This reduction in the number of $(3, 3)$ absorbing set in AB-SC codes compared to AB-LDPC codes in part explains our initial observation in Example 10 where AB-SC codes showed a superior error floor performance compared to AB-LDPC codes.

Lemma 6. *The three VNs in a $(3, 3)$ absorbing sets span at most three consecutive regions.*

Proof. Based on Remark 3 and the fact that each pair of VNs in a $(3, 3)$ absorbing set are connected through a satisfied CN (Figure 5.3), the three VNs span at most three consecutive regions. □

Lemma 6 enables us to categorize all $(3,3)$ absorbing sets in $H(3,p,L,\boldsymbol{\xi})$ into four exhaustive mutually-exclusive cases[5]:

Case 1: All three VNs are in the same region.

Case 2: Two VNs are in the same region, the third VN is in the next region.

Case 3: Two VNs are in the same region, the third VN is in the previous region.

Case 4: The three VNs each belong to a different region and the three regions are consecutive.

(a) Number of absorbing sets in Case 1: In this case, we put a window over each region (R'_1 to R'_{3L+1}) and count the number of $(3,3)$ absorbing sets within that window.

Lemma 7. *The total number of* $(3,3)$ *absorbing sets in Case 1, denoted by* $F_1(p,L,r_1,r_2,r_3,r_4)$*, is:*

$$\begin{aligned} F_1(p,L,r_1,r_2,r_3,r_4) = {} & F_1^{R_1}(p,r_1) + L\cdot F_1^{R_1}(p,r_2) \\ & + (L-1)\cdot F_1^{R_1}(p,r_1+r_4) + L\cdot F_1^{R_1}(p,r_3) + F_1^{R_1}(p,r_4), \end{aligned} \tag{5.6}$$

where $F_1^{R_n}(p,m)$ *is the number of* $(3,3)$ *absorbing sets within region* R_n *of width* m*, for a given circulant size* p*.*

Proof. The total number of $(3,3)$ absorbing set in Case 1 is equal to the summation of the counted absorbing sets within regions R'_1 through R'_{3L+1}, i.e.,

$$F_1(p,L,r_1,r_2,r_3,r_4) = \sum_{n=1}^{3L+1} F_1^{R'_n}(p,r'_n). \tag{5.7}$$

One can show that

$$F_1^{R_1}(p,m) = F_1^{R'_n}(p,m), \quad n\in\{1,\cdots,3L+1\}. \tag{5.8}$$

[5]It is trivial to show that due to the check consistency property of AB-LDPC codes [16], it is not possible to have two VNs in the same region R_n and the third VN in the region R_{n-2} or R_{n+2}.

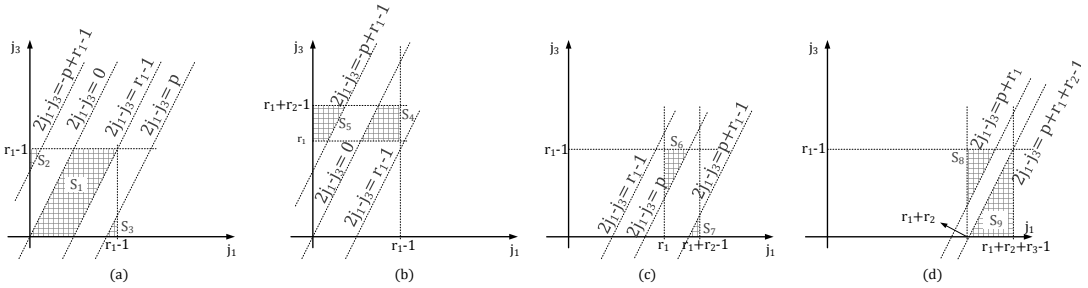


Figure 5.4: (a) Example of Case 1. All VNs in region R'_1. (b) Example of Case 2. VNs 1 and 2 are in region R'_1 and VN 3 is in region R'_2. (c) Example of Case 3. VN 3 is in region R'_1 and VNs 1 and 2 are in region R'_2. (d) Example of Case 4. VN 3 is in region R'_1, VN 2 is in region R'_2 and VN 1 is in region R'_3.

By substituting each term in (5.7) with the LHS of (5.8), and by the fact that for any $k \in \mathbb{N}, 1 \leq k \leq 3L+1$,

$$r'_1 = r_1, \quad \{k : (k \mod 3) = 2\} \rightarrow r'_k = r_2,$$

$$r'_{3L+1} = r_4, \quad \{k : (k \mod 3) = 3\} \rightarrow r'_k = r_3,$$

$$\{k : (k \mod 3) = 1, k \neq 1, k \neq 3L+1\} \rightarrow r'_k = r_1 + r_4,$$

(5.6) can be obtained. □

As an example, we consider that all the VNs are in region R'_1. The problem of finding valid column groups j_1 and j_3 can be graphically interpreted as the problem of counting the integer pairs (j_1, j_3) within the areas S_1, S_2 and S_3 in Figure 5.4(a). Note that based on the values of p and r_1, areas S_2 and S_3 can be either the empty set $\varnothing$ or a triangle. The number of (integer) points existing in S_1, S_2 and S_3, denoted by $N_{S_1}(r_1)$, $N_{S_2}(p, r_1)$ and $N_{S_3}(p, r_1)$, respectively, can be found by (for brevity, details are omitted)

$$N_{S_1}(r_1) = \begin{cases} \frac{(r_1-1)^2}{2} \text{ if } r_1 \text{ is odd}, \\ \frac{r_1(r_1-2)}{2} \text{ if } r_1 \text{ is even}, \end{cases} \quad \text{and} \quad N_{S_2}(p, r_1) = N_{S_3}(p, r_1) = \begin{cases} \frac{(2r_1-p)^2-1}{4} \text{ if } 2r_1 \geq p+2, \\ 0 \text{ if } 2r_1 < p+2. \end{cases}$$

Therefore, the total number of absorbing sets within region R_1 is

$$F_1^{R_1}(p, r_1) = p \cdot \left(N_{S_1}(r_1) + N_{S_2}(p, r_1) + N_{S_3}(p, r_1)\right). \tag{5.9}$$

Note that the multiplication by p in (5.9) is due to p choices for k_1. As an example, if $p = 11$ and $r_1 = 8$, the number of $(3, 3)$ absorbing sets with all their three variable nodes in region 1 is equal to $11 \times (24 + 6 + 6) = 386$.

(b) Number of absorbing sets in Case 2: Here, we put a window over each two consecutive regions ($\{R'_1, R'_2\}$ through $\{R'_6, R'_7\}$).

Lemma 8. *The total number of* $(3, 3)$ *absorbing sets in Case 2 is obtained as follows:*

$$\begin{aligned} F_2(p, L, r_1, r_2, r_3, r_4) = {} & F_2^{R_1}(p, r_1, r_2) + L{\cdot}F_2^{R_2}(p, r_2, r_3) + (L-1){\cdot}F_2^{R_3}(p, r_3, r_1 + r_4) \\ & + (L-1){\cdot}F_2^{R_1}(p, r_1 + r_4, r_2) + F_2^{R_3}(p, r_3, r_4), \end{aligned} \tag{5.10}$$

where $F_2^{R_n}(p, m, k)$ *denotes the number of* $(3, 3)$ *absorbing sets with two VNs in* R_n *of width* m *and one VN in* R_{n+1} *of width* k*.*

A complete discussion of the calculation for each term on the RHS in (5.10) as well as for the related terms $F_3^{R_{1/2/3}}$ and $F_4^{R_{1/2/3}}$ below in Lemma 9 and 10, respectively, can be found in [64].

As an example, the valid areas for the pair (j_1, j_3) when VNs 1 and 2 are in R_1 and VN 3 is in R_2 are shown in Figure 5.4(b). Note that based on the values of p, r_1 and r_2, the areas S_4 and S_5 can be empty set $\varnothing$, a triangle, or a trapezoid.

(c) Number of absorbing sets in Case 3: Similar to Case 2, we put a window over each two consecutive regions. Here, we count the number of $(3, 3)$ absorbing sets which have two VNs in the second region and one VN in the first region.

Lemma 9. *The total number of* $(3, 3)$ *absorbing sets in Case 3 is obtained as follows:*

$$\begin{aligned} F_3(p, L, r_1, r_2, r_3, r_4) = {} & F_3^{R_1}(p, r_1, r_2) + L{\cdot}F_3^{R_2}(p, r_2, r_3) + (L-1){\cdot}F_3^{R_3}(p, r_3, r_1 + r_4) \\ & + (L-1){\cdot}F_3^{R_1}(p, r_1 + r_4, r_2) + F_3^{R_3}(p, r_3, r_4), \end{aligned} \tag{5.11}$$

where $F_3^{R_n}(p, m, k)$ is the number of $(3, 3)$ absorbing sets with one VN in R_n of width m and two VNs in R_{n+1} of width k.

As an example for Case 3, the areas for the pair (j_1, j_3), when VN 3 is in R_1 and VNs 2 and 3 are in R_2, are displayed in Figure 5.4(c). Again based on the values of p, r_1, and r_2, areas S_6 and S_7 can be $\varnothing$, a triangle, or a trapezoid.

(d) Number of absorbing sets in Case 4: Here, we put a window over each three consecutive regions. For each window, we count the number of $(3, 3)$ absorbing sets which have one VNs in each region.

Lemma 10. *The total number of $(3, 3)$ absorbing sets in Case 4 is obtained as follows:*

$$\begin{aligned} F_4(p, L, r_1, r_2, r_3, r_4) = (L-1)\cdot F_4^{R_2}(p, r_2, r_3, r_1 + r_4) + F_4^{R_1}(p, r_1, r_2, r_3) + F_4^{R_2}(p, r_2, r_3, r_4) \\ + (L-1)\cdot F_4^{R_3}(p, r_3, r_1 + r_4, r_2) + (L-1)\cdot F_4^{R_1}(p, r_1 + r_4, r_2, r_3), \end{aligned} \tag{5.12}$$

where $F_4^{R_n}(p, m, k, \ell)$ is the number of $(3, 3)$ absorbing sets with one VN in R_n of width m, one VN in R_{n+1} of width k and one VN in R_{n+2} of width ℓ.

As an example, Figure 5.4(d) highlights the valid areas for the pair (j_1, j_3) when VN 3 is in R_1, VN 2 is in R_2, and VN 1 is in R_3. Again based on the values of p, r_1, r_2 and r_3, the areas S_8 and S_9 can be $\varnothing$, a triangle or a trapezoid.

For a given circulant size p, a coupling length L and a cutting vector $\boldsymbol{\xi}$, the following equation provides the exact number of $(3, 3)$ absorbing sets, which is denoted by $A_{(3,3)}(3, p, L, \boldsymbol{\xi})$:

$$A_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \sum_{n=1}^{4} F_n(p, L, r_1, r_2, r_3, r_4), \tag{5.13}$$

where $r_1 = \xi_1$, $r_2 = \xi_2 - \xi_1$, $r_3 = \xi_3 - \xi_2$ and $r_4 = p - \xi_3$, and the functions $F_n(p, L, r_1, r_2, r_3, r_4), n = \{1, 2, 3, 4\}$, can be calculated as in (5.6), (5.10), (5.11), and (5.12).

Remark 4. *The absorbing set enumeration method presented in this section can be applied to any* (a, b) *absorbing set. However, for larger absorbing sets, the problem is more involved as the problem of finding the valid points for the column group indices is over higher dimensional spaces. For larger absorbing sets, column group indices can not necessarily be specified as a function of only two column groups indices. Therefore, the areas for the valid column groups must be described over higher dimensional spaces.*

5.4.1.2 Analysis of $(4, 2)$ absorbing sets

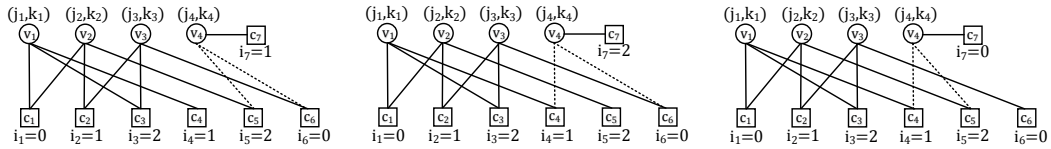
In this section, we provide an approach to find the exact number of $(4, 2)$ absorbing sets in AB-SC codes with column weight 3 based on the number of $(3, 3)$ absorbing sets presented in the previous section. These results further explain the differences in both the error profile spectrum and the performance between AB-LDPC and AB-SC codes observed in Example 10.

It was shown in [16] that (in the block case) each $(4, 2)$ absorbing sets is formed by exactly two distinct $(3, 3)$ absorbing sets. We first review how to count the number of $(4, 2)$ absorbing sets in AB-LDPC codes [16].

Lemma 11. *(*[16]*) The total number of* $(4, 2)$ *absorbing sets in the Tanner graph corresponding to* $H(3, p)$ *is equal to* $\frac{3p^2(p-1)}{2}$.

Proof. Each $(4, 2)$ absorbing sets is formed by connecting a pair of unsatisfied check nodes in a $(3, 3)$ absorbing set by a new check node. Since each $(3, 3)$ absorbing set has 3 unsatisfied check nodes, this results in $\binom{3}{2} = 3$ size $(4, 2)$ absorbing sets. □

Note that in the case of AB-LDPC codes, as discussed in the proof above, each $(3, 3)$ absorbing set always leads to three $(4, 2)$ absorbing sets. In other words, for each two chosen unsatisfied check nodes, there always exists a new variable node to satisfy these check nodes in a $(4, 2)$ absorbing set. This is illustrated in Figure 5.5 which



(a) (b) (c)

Figure 5.5: The three possible cases for $(4, 2)$ absorbing sets.

shows the three possible cases for $(4, 2)$ absorbing sets. In contrast, in the case of AB-SC codes, due to the structure of these codes, one may not always be able to find a new variable node connected to each pair of unsatisfied check nodes in a $(3, 3)$ absorbing set to complete a $(4, 2)$ absorbing set configuration.

We first categorize $(4, 2)$ absorbing sets into three cases based on the row group indices of the satisfied check nodes connected to variable node v_4:

- **Case (a):** v_4 is connected to satisfied check nodes with $i = 0$ and $i = 2$ (Figure 5.5(a)).
- **Case (b):** v_4 is connected to satisfied check nodes with $i = 0$ and $i = 1$ (Figure 5.5(b)).
- **Case (c):** v_4 is connected to satisfied check nodes with $i = 1$ and $i = 2$ (Figure 5.5(c)).

In addition to the conditions in (5.4), each case above imposes a separate new pattern consistency condition. Case (a) imposes that $k_3 \equiv k_4 \mod p$ and $k_2 + 2j_2 \equiv k_4 + 2j_4 \mod p$, where k_2, k_3, k_4, j_2, and j_4 can be any integer in $\{0, 1, \cdots, p-1\}$. These conditions lead to

$$j_1 \equiv j_4 \mod p. \tag{5.14}$$

The configuration in Case (b) results in the equations $k_3 \equiv k_4 \mod p$ and $k_2 + 2j_2 \equiv k_4 + 2j_4 \mod p$ which lead to

$$2j_3 - j_1 \equiv j_4 \mod p. \tag{5.15}$$

Similarly, the pattern consistency in Case (c) leads to $k_2 + 2j_2 \equiv k_4 + 2j_4 \mod p$ and $k_1 + j_1 \equiv k_4 + j_4 \mod p$, that result in

$$3j_1 - 2j_3 \equiv j_4 \mod p. \tag{5.16}$$

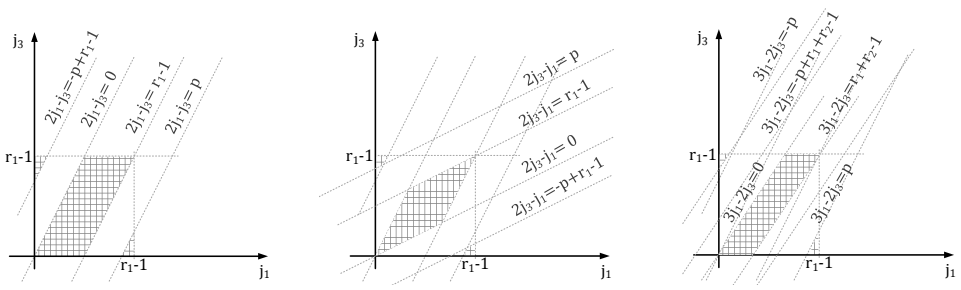
In the previous subsection, the problem of counting $(3, 3)$ absorbing sets was mapped into a problem of counting the number of valid points for the pair (j_1, j_3) in a two dimensional space. Here, we apply a similar approach to count the number of $(4, 2)$ absorbing sets. Equations (5.14), (5.15), and (5.16) further limit the areas for valid choices of (j_1, j_3) pairs in the two dimensional space. The following example clarifies the procedure for counting the number of $(4, 2)$ absorbing sets in AB-SC codes based on the previously counted $(3, 3)$ absorbing sets.

Example 12. *Consider* $(3, 3)$ *absorbing sets with* v_1*,* v_2*, and* v_3 *all in region* R_1*. The valid area for the choice of the pair* (j_1, j_3) *in this case is shown in Figure 5.4(a) and the count of these absorbing sets is expressed as* $F_1^{R_1}(p, r_1)$ *in* (5.9)*. Based on* (5.14)*,* (5.15)*, and* (5.16)*, we find the areas for the valid choices of pairs* (j_1, j_3) *in the three* $(4, 2)$ *absorbing set cases discussed above, shown in Figure 5.6.*

- ***Case (a):*** v_4 *is connected to the two unsatisfied check nodes of* $(3, 3)$ *absorbing set with* $i = 0$ *and* $i = 2$*. This condition imposes that* v_4 *be in the region* R_1 *(i.e.,* $0 \leq j_4 \leq r_1 - 1$*). As shown in* (5.14)*,* $j_4 = j_1$ *and* $k_4 = k_3$*. Since* $0 \leq j_4 \leq r_1 - 1$*, the addition of variable node* v_4 *does not impose any new constraints of the choice of the pair* (j_1, j_3)*, as shown in Figure 5.6(a).*

- ***Case (b):*** *Since* v_4 *is connected to the unsatisfied check node of* $(3, 3)$ *absorbing set with* $i = 0$*, it must lie again within* R_1*. Therefore, we obtain* $0 \leq j_4 \leq r_1 - 1$*, and based on* (5.15)*,* $0 \leq 2j_3 - j_1 \leq r_1 - 1 \mod p$*. This inequality imposes new constraints on the allowable choices for the pair* (j_1, j_3)*, as illustrated in Figure 5.6(b).*

- ***Case (c):*** *In this case,* v_4 *belongs to either* R_1 *or* R_2*. Thus,* $0 \leq j_4 \leq r_1 + r_2 - 1$*.*



(a) (b) (c)

Figure 5.6: Areas corresponding to valid choices for j_1 and j_3 in Example 12.

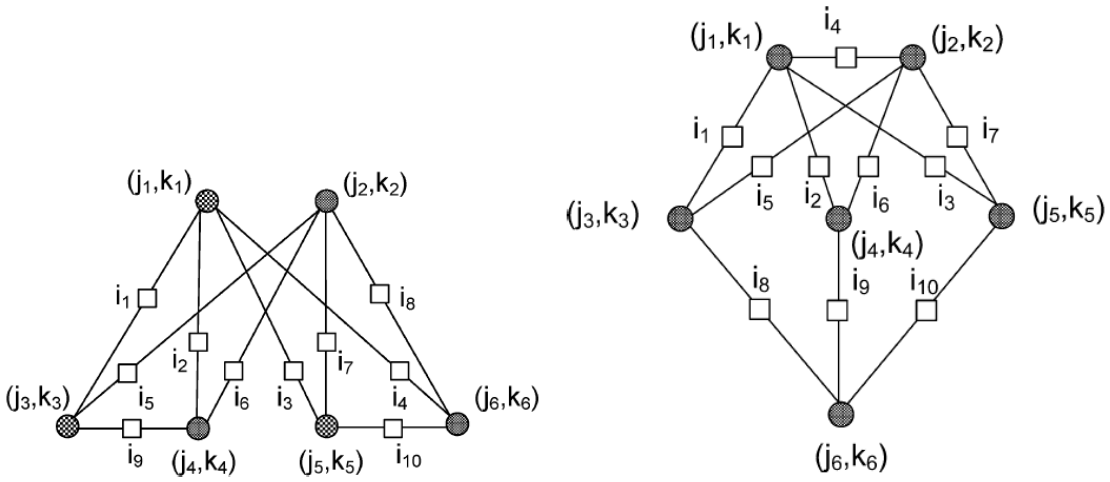
Using (5.15), $0 \leq 3j_1 - 2j_3 \leq r_1 + r_2 - 1 \mod p$. *This inequality imposes additional constraints on the choice of* (j_1, j_3)*, as shown in Figure 5.6(c).*

In the above example, we have only considered $(3, 3)$ absorbing sets with all their variable nodes in region R_1 and we have computed the number of the resultant $(4, 2)$ absorbing sets. To enumerate all $(4, 2)$ absorbing sets, we need to consider each term in Lemmas 5, 6, 7, and 8 one by one. A detailed discussion on valid choices for pairs (j_1, j_3) for each term in Lemmas 5, 6, 7, and 8 can be found in [64].

Remark 5. *Note that each* $(3, 3)$ *absorbing set in AB-LDPC codes results in three* $(4, 2)$ *absorbing sets whereas each* $(3, 3)$ *absorbing set in AB-SC codes does not necessarily result in three* $(4, 2)$ *absorbing sets. The reason is that in the AB-SC case, the valid areas for the pairs* (j_1, j_3) *in the cases (a), (b), and (c) above are always subareas of the valid area for* $(3, 3)$ *absorbing sets. This is demonstrated in Example 12 and explains our observation in Example 10 where, compared to AB-LDPC codes, a smaller percentage of errors in the error profile of AB-SC codes is due to* $(4, 2)$ *absorbing sets.*

5.4.2 Column Weights 4 Analysis

In this section, we provide analysis of the minimal absorbing sets for $c = 4$ AB-SC codes.



(a) Configuration A (b) Configuration B

Figure 5.7: Two candidate configurations for $(6, 4)$ absorbing sets.

Lemma 12. *The smallest absorbing set possible in Tanner graph of AB-SC codes with column weight 4 is a* $(6, 4)$ *absorbing set.*

Proof. The proof follows immediately from the arguments in [16]. □

As shown in [16], there are two possible configurations for $(6, 4)$ absorbing sets in AB-LDPC codes, shown in Figures 5.7(a), 5.7(b). In this section, we will present the derivations for Configuration A shown in Figure 5.7(a). A similar approach can be used for the analysis of Configuration B. Details can be found in [64].

Based on the bit consistency condition, it is shown in [16] that there are only two distinct labelings possible for row groups of the check nodes in this configuration:

$$(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}) \in \{(x, y, z, w, y, x, w, z, z, y), (x, y, z, w, y, z, w, x, w, y)\}, \tag{5.17}$$

where w, x, y, and z are distinct row group indices in $\{0, 1, 2, 3\}$. However, the labeling $(x, y, z, w, y, z, w, x, w, y)$ do not lead to valid solutions for $p \notin \{2, 3, 5, 7, 37\}$ as shown in [16]. Therefore, we only consider the labeling $(x, y, z, w, y, x, w, z, z, y)$ in

Table 5.2: Valid parameters for the Configuration A for the case of $z = 0$.

x, y, w	j_1	j_2	j_3	j_4	j_5	j_6	k_1	k_2	k_3	k_4	k_5	k_6
$1, 3, 2$	ρ	$\rho + 4\tau$	$\rho + 3\tau$	$\rho + \tau$	$\rho + \tau$	$\rho + 3\tau$	δ	$\delta - 6\tau$	$\delta - 3\tau$	$\delta - 3\tau$	δ	$\delta - 6\tau$

Table 5.3: Valid parameters for the Configuration A for the case of $x = 0$.

y, z, w	j_1	j_2	j_3	j_4	j_5	j_6	k_1	k_2	k_3	k_4	k_5	k_6
$2, 1, 3$	ρ	ρ	$\rho - \tau$	$\rho + \tau$	$\rho - \tau$	$\rho + \tau$	δ	$\delta - 2\tau$	δ	$\delta - 2\tau$	$\delta + \tau$	$\delta - 3\tau$

our discussion. By symmetry, it can be shown that all valid choices for row group indices can be categorized into (1) Case $z = 0$ and (2) Case $x = 0$. By applying the patter consistency, it is shown in [16] that the only valid choices for column group indices j_1 through j_6 and also k_1 through k_6 are the ones shown in Tables 5.2 and 5.3.

The structure of AB-SC codes imposes additional constraints on the choice of column group indices in $(6, 4)$ absorbing sets, as shown in the following Lemma.

Lemma 13. *The six VNs of* $(6, 4)$ *absorbing sets in an AB-SC code span at most seven consecutive regions.*

Proof. Our proof is based on contradiction. Assume that the six VNs in a $(6, 4)$ absorbing set span eight consecutive regions, namely R_1 through R_8. Based on the structure of AB-SC codes, two VNs sharing a satisfied CN must belong to a window of at most $c = 4$ consecutive regions. For instance, suppose that $j_1 \in R_1$ and $j_2 \in R_8$. According to the configuration, j_1 and j_3 share a satisfied CN, therefore j_3 must belong to one of the regions R_1, R_2, R_3, or R_4. VN j_3 also shares a satisfied CN with j_2, thus, j_3 must belong to one of regions R_5, R_6, R_7, or R_8, which is in contradiction with the former argument. The same logic holds for choices other than j_1 and j_2 for the first and last regions. □

Based on the fact that the six VNs of the $(6, 4)$ absorbing set, $\{j_1, j_2, j_3, j_4, j_5, j_6\}$, span at most seven consecutive regions, the following cases must be considered in the enumeration of $(6, 4)$ absorbing sets.

- **Case 1:** All the six VNs are in the same region.
- **Case 2:** The six VNs span two consecutive regions.
- **Case 3:** The six VNs span three consecutive regions.
- **Case 4:** The six VNs span four consecutive regions.
- **Case 5:** The six VNs span five consecutive regions.
- **Case 6:** The six VNs span six consecutive regions.
- **Case 7:** The six VNs span seven consecutive regions.

For brevity of the discussion we consider Case 1 and illustrate the counting procedure with an example. A similar procedure can be applied to other JKcases, where details can be found in [64].

Example 13. *In this example, we provide the exact number of* $(6, 4)$ *absorbing sets with Configuration A (shown in Figure 5.7(a)), when all variable nodes are within region* R_1*. In the case that* $z = 0$*, the valid choices for the column group indices are given in Table 5.2. These valid column group assignments impose that* $4j_3 = 4j_6 = 3j_2 - j_1$ *and* $4j_4 = 4j_5 = 3j_1 + j_2$*. Since* $0 \leq j_1, j_2, j_3, j_4, j_5, j_6 \leq r_1 - 1$*, the following inequalities must be satisfied for the choice of the pair* (j_1, j_2)*.*

$$
\begin{aligned}
&0 \leq j_1 \leq r_1 - 1, \quad 0 \leq j_2 \leq r_1 - 1, \\
&0 \leq \frac{1}{4}(3j_2 - j_1) \leq r_1 - 1 \mod p, \\
&0 \leq \frac{1}{4}(3j_1 + j_2) \leq r_1 - 1 \mod p.
\end{aligned}
$$

The valid area corresponding to the above inequalities is shown in Figure 5.8(a). Similarly, for the case that $x = 0$*, we have* $j_2 = j_1$*,* $j_3 = j_6 = 2j_1 - j_3$*. The valid areas for the pair* (j_1, j_3) *in this case is shown in Figure 5.8(b). As a numerical example, assume*

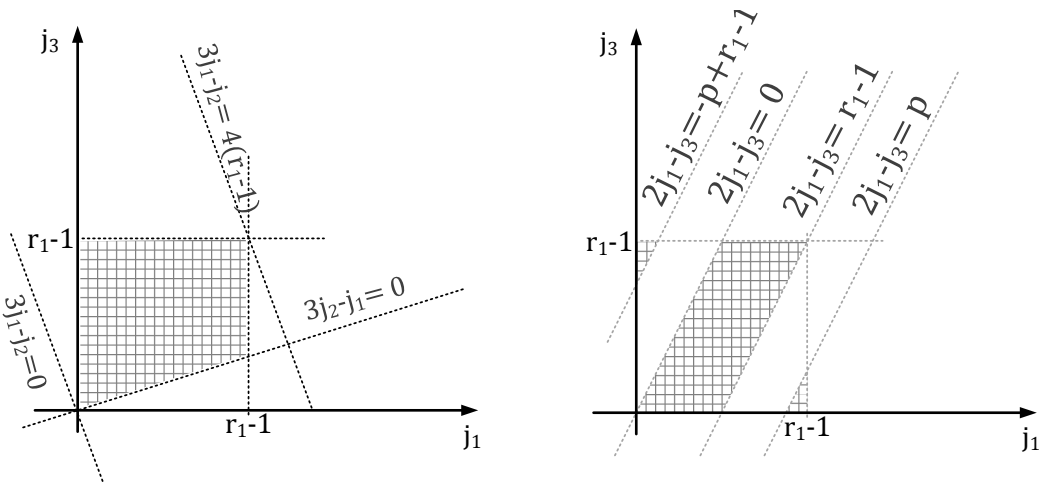


Figure 5.8: Valid areas for $(6, 4)$ absorbing set with Configuration A, when all variable nodes are in region R_1.

that $p = 53$ and the width of region R_1 is $r_1 = 18$. Then, the number of integer points in the valid areas shown in Figures 5.8(a) and 5.8(b) are 82 and 162, respectively. Since there exist $p = 53$ choices for k_1, in this example, the total number of $(6, 4)$ absorbing sets with Configuration A when all variable nodes are within region R_1 is equal to $53 \cdot (162 + 82) = 12932$.

5.5 Optimal Cutting Vector for AB-SC Codes

Based on our analysis in Section 5.4, we provide the optimal choice of the cutting vector for AB-SC codes.

Definition 2. ([63]) *For any given prime circulant size p, a optimal cutting vector corresponding to absorbing set (a, b) is defined as $\boldsymbol{\xi}^*_{(a,b)}(p) = \arg\min_{\boldsymbol{\xi}} \lim_{L\to\infty} A_{(a,b)}(c, p, L, \boldsymbol{\xi})$.*

Theorem 7. *An optimal cutting vector for $(3, 3)$ absorbing sets in column weight 3*

Table 5.4: Optimal cutting vectors for various circulant sizes.

p	Optimal cutting vector
67	[15, 31, 47]
97	[23, 48, 73]
107	[28, 54, 81]
113	[29, 57, 88]

AB-SC codes with circulant size p is

$$\begin{aligned}\boldsymbol{\xi}^*_{(a,b)}(p) = \arg\min_{\boldsymbol{\xi}}\{&F_1^{R_1}(p, r_2) + F_1^{R_1}(p, r_1 + r_4) + F_1^{R_1}(p, r_3) \\ &+ F_2^{R_2}(p, r_2, r_3) + F_2^{R_3}(p, r_3, r_1 + r_4) + F_2^{R_1}(p, r_1 + r_4, r_2) + F_3^{R_2}(p, r_2, r_3) \\ &+ F_3^{R_3}(p, r_3, r_1 + r_4) + F_3^{R_1}(p, r_1 + r_4, r_2) + F_4^{R_2}(p, r_2, r_3, r_1 + r_4) \\ &+ F_4^{R_3}(p, r_3, r_1 + r_4, r_2) + F_4^{R_1}(p, r_1 + r_4, r_2, r_3)\}, \end{aligned} \tag{5.18}$$

where $r_1 = \xi_1$, $r_2 = \xi_2 - \xi_1$, $r_3 = \xi_3 - \xi_2$ and $r_4 = p - \xi_3$ and the functions on the RHS of (5.18) *are shown in Lemma 7, 8, 9, and 10.*

Proof. As $L \to \infty$, the terms in (5.6), (5.10), (5.11), and (5.12) grow linearly with L and therefore become dominant terms. By summing these dominant terms we obtain the RHS of (5.18). □

Remark 6. *Based on Theorem 7, the problem of finding the optimal cutting vector is essentially mapped to finding an integer vector which minimizes the function in* (5.18). *Compared to [63], where the optimal cutting vector for $p \leq 23$ is found by an exhaustive computer search, our approach is computationally less complex and offers the optimal cutting vector for large choices of p. As an example, a list of optimal cutting vectors for $p = 67, 97, 107, 113$ is provided in Table 5.4.*

Our experimental results in Section 5.7 show that binary AB-SC codes with the optimal cutting vector provide a performance improvement by more than one order of magnitude compared to AB-SC codes with randomly chosen cutting vectors.

Although the choice of cutting vector significantly affects the performance of binary AB-SC codes, we show in Section 5.6 that the choice of the cutting vector is not a critical parameter in optimizing non-binary AB-SC codes since the non-binary edge weights provide enough degree of freedom to remove problematic absorbing sets from non-binary AB-SC codes.

5.6 Absorbing Set Analysis for Non-Binary AB-SC Codes

It was recently shown that non-binary spatially-coupled codes have superior iterative threshold and finite-length performance compared to non-binary block LDPC codes [65–67]. Motivated by these findings, in this section we provide a study of non-binary absorbing sets for non-binary AB-SC codes. In particular, we first present the average number of $(3, 3)$ non-binary absorbing sets in non-binary AB-SC codes. We also provide an approach to design non-binary AB-SC codes with a reduced number of problematic non-binary absorbing sets.

The following theorem provides the average number of $(3, 3)$ non-binary absorbing sets in non-binary AB-SC codes with column weight 3.

Theorem 8. *Consider an AB-SC code over GF(q), where q is a power of a prime number, with circulant size p, coupling length L, cutting vector $\boldsymbol{\xi}$, and random assignment of edge weights. The number of $(3, 3)$ non-binary absorbing sets, averaged over all possible edge weight assignments, denoted by $\overline{A}^{q}_{(3,3)}(3, p, L, \boldsymbol{\xi})$, is*

$$\overline{A}^{q}_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \frac{1}{(q-1)} \sum_{n=1}^{4} F_n(p, L, r_1, r_2, r_3, r_4), \tag{5.19}$$

where $r_1 = \xi_1$, $r_2 = \xi_2 - \xi_1$, $r_3 = \xi_3 - \xi_2$ and $r_4 = p - \xi_3$, and the functions $F_n(p, L, r_1, r_2, r_3, r_4), n = \{1, 2, 3, 4\}$, can be calculated as in (5.6)*,* (5.10)*,* (5.11)*, and* (5.12)*.*

Proof. Consider the structure of an $(3, 3)$ absorbing set in Figure 5.3 with non-binary edge weights $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_6$. There exist $(q-1)^6$ unique choices for a set of six

edge weights. Based on the weight condition of non-binary absorbing sets in (3.5), the edge weights in a non-binary absorbing set satisfy $\mathrm{w}_1\mathrm{w}_3\mathrm{w}_5 = \mathrm{w}_2\mathrm{w}_4\mathrm{w}_6 \mod q$. By choosing w_1 through w_5 independently from $GF(q)\backslash 0$, the edge weight w_6 can be uniquely determined. Thus, there exist $(q-1)^5$ choices for edge weights which result in absorbing sets over GF(q). As a result, the average number of $(3,3)$ non-binary absorbing sets is

$$\overline{A}^{q}_{(3,3)}(3,p,L,\boldsymbol{\xi}) = \frac{(q-1)^5}{(q-1)^6} A_{(3,3)}(3,p,L,\boldsymbol{\xi}) = \frac{1}{(q-1)} \sum_{n=1}^{4} F_n(p,L,r_1,r_2,r_3,r_4).$$

□

The above theorem illustrates that on average, a ratio of $\frac{1}{(q-1)}$ of all binary $(3,3)$ absorbing sets in the unlabeled Tanner graph of a non-binary AB-SC code result in problematic non-binary absorbing sets. As an example, consider $q = 3$. Then, on average, 50% of the binary absorbing sets in the unlabeled Tanner graph do not result in problematic substructures in the Tanner graph after a random assignment of edge weights over GF(3). This observation in part explains the better performance of non-binary AB-SC codes in error floor region compared to their binary counterparts.

Our results in Section 5.7 show that our optimized NB AB-SC have performance improvements of more than one order of magnitude compared to unoptimized[6] non-binary AB-SC codes.

Remark 7. *In Section 5.4, we have observed that in the case of binary AB-SC codes the number of binary absorbing sets is significantly affected by the choice of the cutting vector. In contrast to binary AB-SC codes, the choice of the cutting vector in non-binary AB-SC codes is not as critical since, irrespective of the choice of the cutting vector, the non-binary edge weights offer enough degree of freedom to remove problematic small absorbing sets. For a given underlying AB-LDPC code, different choices of the cutting*

[6]The unoptimized AB-SC are constructed by randomly assigning non-binary edge weights to an unlabeled AB Tanner graph.

Table 5.5: $(3, 3)$ absorbing set comparison for various cutting vectors, $p = 67$, $L = 2$, $q = 2$.

Optimal	Random ([10, 18, 56])	Average
273561	379957	441293

vector result in $\binom{p}{c}$ different binary AB-SC codes. For a given underlying binary AB-LDPC code, both a fixed cutting vector and field size q, there are $(q-1)^{cp^2}$ different non-binary AB-SC codes (obtained by different edge weight selections). Since generally $(q-1)^{cp^2} >> \binom{p}{c}$, this roughly verifies the significantly larger degrees of freedom offered by the choice of the edge weights compared to the ones offered by the choice of the cutting vector.

5.7 Experimental Results

In this section, we present the simulation results for our designed binary and non-binary AB-SC codes. In the binary case we show that the choice of the optimal cutting vector in the design of AB-SC codes results in fewest binary absorbing sets, and as a result, in a superior error floor performance. In the non-binary case, our results demonstrate that our algorithm discussed in Section 5.6 significantly reduces the non-binary absorbing sets. Therefore, our optimized non-binary AB-SC codes have better error floor performance compared to non-binary AB-SC codes with randomly assigned edge weights.

Figure 5.9 shows the performance comparison between binary AB-SC codes with constraint length $\nu_s = 4489$, $p = 67$ and $L = 2$, constructed using cutting vectors $\boldsymbol{\xi}^*_{(3,3)}$ (optimal for $(3, 3)$ absorbing sets), $\boldsymbol{\xi}^*_{(4,2)}$ (optimal for $(4, 2)$ absorbing sets) and a randomly selected cutting vector $\boldsymbol{\xi} = [10, 18, 56]$, when decoded using a sliding window decoder implemented based on the soft-xor [35] algorithm. The code with cutting vector $\boldsymbol{\xi}^*_{(3,3)}$ shows about one order of magnitude performance improvement in the error

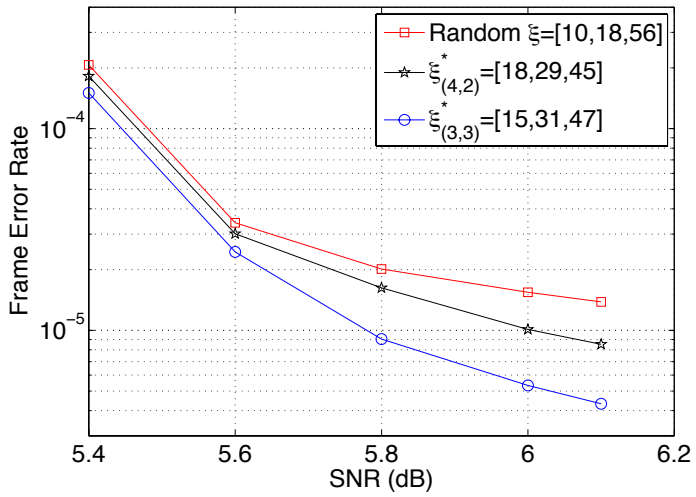


Figure 5.9: Performance comparison for binary SC codes constructed by various cutting vectors.

floor region, compared to the code with the random cutting vector. Furthermore, the code with cutting vector $\boldsymbol{\xi}^*_{(3,3)}$ performs significantly better that the one with $\boldsymbol{\xi}^*_{(4,2)}$. This is consistent with our initial observation in Example 1, where the dominant absorbing set in error profile of the AB-SC code was $(3,3)$, and the number of $(4,2)$ absorbing sets was negligible. Therefore, it is clear that the code optimized to minimize $(3,3)$ absorbing sets performs better in the error floor region.

Table 5.5 compares the number of $(3,3)$ binary absorbing sets for the codes shown in Figure 5.9. Table 5.5 also includes the average number of $(3,3)$ absorbing sets over all possible selections of cutting vectors for $H(3,p,L,\boldsymbol{\xi})$. It can be observed that the optimal cutting vector reduces the number of $(3,3)$ absorbing sets by approximately 39% compared to the average number of absorbing sets over all possible cuts.

For the binary AB-SC codes, our code design is limited to the choice of the cutting vector, once the underlying block code is specified. In the non-binary case, the choice of the edge weights offers significantly more degrees of freedom that can be exploited in an optimized code design. In fact, our results show that the cutting vector choice is not a critical parameter in the design of non-binary AB-SC codes since we can remove

all non-binary absorbing sets of interest only by manipulating the edge weights.

Figure 5.10 shows the simulation results for non-binary AB-SC codes over GF(4) with constraint length $\nu_s = 3698$ bits. To construct the non-binary codes, we first form binary AB-SC parity-check matrices with $p = 43$ and column weight 3 using the optimal cutting vector ($\boldsymbol{\xi}_1 = [10, 22, 34]$) and a randomly chosen cutting vector ($\boldsymbol{\xi}_2 = [5, 29, 33]$). Then, the unoptimized non-binary AB-SC codes are constructed by randomly assigning non-binary edge weights on top of the unlabeled AB-SC Tanner graphs. The optimized codes are constructed by manipulating the edge weights (based on the algorithm addressed in Section 5.6) such that $(3, 3)$, $(4, 2)$, and $(4, 4)$ non-binary absorbing sets are completely removed from the Tanner graphs of the original codes. Figure 5.10 shows that our optimized codes have a performance improvement of more than one order of magnitude compared to the original codes. Furthermore, we observe that the performance of the optimized code with a randomly chosen cutting vector is very close to the optimized code with the optimal cutting vector. This observation suggests that the choice of the cutting vector does not significantly affect the performance of our designed non-binary AB-SC codes. The non-binary edge weight offers enough degrees of freedom in codes design such that all problematic small non-binary absorbing set can be removed from the designed codes, regardless of the choice of the cutting vector. In the same figure, it is shown that the unoptimized AB-SC code over GF(3) and GF(4) has better FER performances compared to the binary AB-SC code. An interesting observation is the improved FER performance of the optimized AB-SC code over GF(3) compared to the unoptimized AB-SC code over GF(4), which in combination with its lower decoder complexity, can make the optimized AB-SC code over GF(3) a very attractive candidate for practical implementation. The performance of a binary AB-SC code with $\nu_s = 3721$, $p = 61$ constructed using optimal cutting vector for $(3, 3)$ absorbing set $\boldsymbol{\xi}^*_{(3,3)} = [14, 29, 46]$ is also plotted in Figure 5.10. It can be observed that although the constraint length of the binary AB-SC code and non-binary AB-SC codes are similar, the non-binary codes have superior performance.

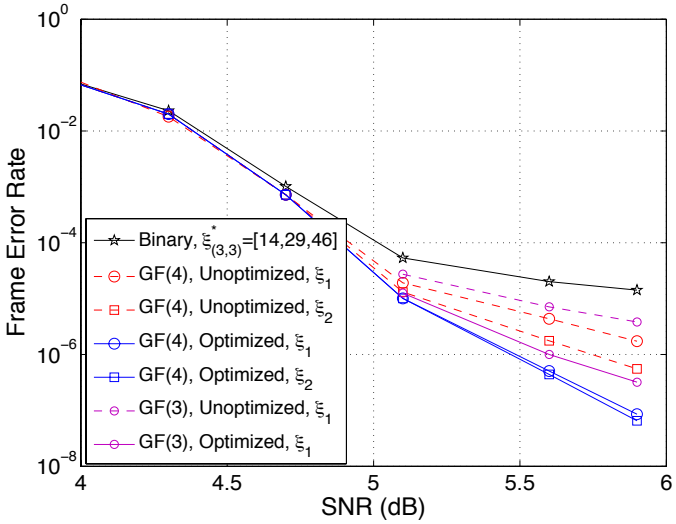


Figure 5.10: Performance comparison for optimized and unoptimized non-binary AB-SC codes over GF(3) and GF(4) with $p = 43$, column weight 3 and cutting vectors $\boldsymbol{\xi}_1 = [10, 22, 34]$ and $\boldsymbol{\xi}_2 = [5, 29, 33]$, where $\boldsymbol{\xi}_1$ is optimized and $\boldsymbol{\xi}_2$ selected randomly, respectively. The binary AB-SC code has constraint length $\nu_s = 3721$, and is constructed using optimal cut for $(3, 3)$ absorbing sets.

CHAPTER 6

Summary of Results

In this dissertation, we provided new combinatorial-based insight into analysis and design of non-binary graph-based codes.

We first started by providing finite-length analysis of non-binary graph-based codes. We introduced a generalized absorbing set definition for non-binary graph-based codes over GF(q). We observed that for non-binary codes where each edge in the Tanner graph has a weight chosen from non-zero elements of GF(q), not only the edges of absorbing sets are topologically connected in specific ways, but also their weights must satisfy certain conditions. We showed that in the case of non-binary elementary absorbing sets, the weight conditions can be simplified. Further, depending on the variable node input of a subgraph which satisfies the non-binary absorbing set conditions, the subgraph may or may not result in an absorbing set error (i.e., the error is input dependent). We proposed an algorithm to decrease the number of absorbing sets in the Tanner graph by only changing carefully chosen edge weights while the topology of the Tanner graph remains the same. Our simulation results using different code parameters and code structures as well as using different decoders confirmed the effectiveness of our proposed approach. Using techniques from graph theory, we showed that as field size gets larger, it is harder to satisfy the edge labeling conditions for absorbing sets and as a result the number of absorbing sets decreases for larger field sizes.

Next, we presented the asymptotic analysis for binary and non-binary regular LDPC codes. We computed the normalized logarithmic asymptotic distributions of absorbing sets and fully absorbing sets for regular LDPC code ensembles. The results are

based on the approach for the trapping set analysis previously proposed by Milenkovic. We also derived simplified formulas for enumerating elementary absorbing sets of the $(3, r)$ and $(4, r)$ LDPC code ensembles. By fixing different graph-theoretic parameters, we analyzed different sizes of (elementary) trapping sets, elementary absorbing sets, and elementary fully absorbing sets of random LDPC code ensembles. These results quantify discrepancies among trapping sets, absorbing sets and fully absorbing sets. Our results show that for small θ and λ, when the rate is moderate, absorbing sets are approximately fully absorbing sets, but that the discrepancy increases with rate. We showed that denser graphs of the prescribed code rate can prevent certain absorbing sets to exist altogether. Also, trapping sets are approximately absorbing sets when θ and λ are small and when the ratio λ is smaller θ than 1. As the ratio approaches 1, the discrepancy between trapping and absorbing sets increases dramatically. Such an observation implies that a trapping set enumeration alone may not give a good indication of error floor under practical (bit-flipping like) decoders. Comparison with known results on trapping sets of some popular structured LDPC code ensembles suggests that these codes asymptotically possess good absorbing set properties. We also computed the normalized logarithmic asymptotic distributions of non-binary absorbing sets for regular $(3, r)$ and $(4, r)$ LDPC code ensembles. We observed that only a small fraction of unlabeled trapping sets leads to problematic non-binary absorbing sets; this results in non-binary absorbing set enumerators providing a better assessment of the error floor compared to trapping set enumerators.

We also looked at a family of structured LDPC codes which are constructed based on protographs. We first presented a class of structured non-binary LDPC codes built out of protographs, called NBPB codes, wherein we considered unconstrained edge weight selections. Equipped with combinatorial definitions extended to the non-binary domain, ensemble enumerators of codewords and trapping sets are calculated. The exact enumerators are presented in the finite-length regime, and the corresponding growth rates are calculated in the asymptotic regime. In many instances, non-binary construc-

tions were shown to have superior properties compared to their binary counterparts. The results presented provide a new analytical toolbox for analysis of non-binary graph-based codes.

We also studied graph-based codes which are constructed by spatially-coupling of LDPC block codes. In particular, we provided a detailed mathematical analysis for the finite-length performance of both binary and non-binary array-based spatially-coupled codes. We presented an approach to enumerate the problematic absorbing sets in array-based spatially-coupled codes. We graphically mapped the absorbing set enumeration problem to a problem of counting the number of valid integer points inside an area in 2-dimensional space. Using this method, the exact number of smallest absorbing sets in column weight 3 and 4 binary array-based spatially-coupled codes was presented as a function of the code parameters (circulant size, coupling length, and cutting vector). Based on our absorbing set enumeration, we found the optimal cutting vector, which minimizes the number of absorbing sets for any choice of the circulant size in binary array-based spatially-coupled codes. In our example, we showed that, compared to the average number of binary absorbing sets over all possible cutting vectors, choosing the minimum cutting vector results in a significant reduction of the number of absorbing sets. In the case of non-binary spatially-coupled codes, we first provided the average number of $(3, 3)$ absorbing sets in array-based spatially-coupled codes over GF(q). We showed that on average, only ratio of $\frac{1}{q-1}$ of binary absorbing sets in unlabeled Tanner graph result in problematic non-binary absorbing sets, when we use a random assignment of non-binary edge weights. This analysis in part explains the superior error floor performance of non-binary spatially-coupled codes compared to their binary counterparts. We also provided an optimization algorithm for non-binary code design. By manipulation of selected edge weights in the Tanner graph of array-based spatially-coupled codes, our optimization algorithm further reduced the number of non-binary absorbing sets in the Tanner graph of non-binary array-based codes. Simulation results confirmed the improved performance for our designed both binary and non-binary

spatially-coupled codes.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication,"*The Bell Systems Technical Journal*, vol. 27, pp. 379 - 423, 623 - 656, Oct. 1948.

[2] P. Elias, "Coding over noisy channels," *IRE International Convention Record*, vol. 3, no. 4, pp. 37 - 46, 1955.

[3] R.W. Hamming, "Error detecting and error correcting codes," *The Bell Systems Technical Journal*, vol. 29, no. 2, pp. 147 - 150, Apr. 1950.

[4] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68 - 79, Mar. 1960.

[5] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300 - 304, Jun. 1960.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding," in *Proceedings of the IEEE International Conference on Communications*, pp. 1064 - 1070, May 1993.

[7] R. G. Gallager. *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[8] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronic Letters*, vol. 32, p. 1645, Aug. 1996.

[9] Y. Y. Tai, L. Lan, L. Zeng, S. Lin, and K. Abdel-Ghaffar, "Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels," *IEEE Trans. Info. Theory*, vol. 54(10), pp. 1765 - 1774, Oct. 2006.

[10] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165-167, Jun. 1998.

[11] J. Kang, Q. Huang, L. Zhang, B. Zhou, and S. Lin, "Quasi-cyclic LDPC codes: an algebraic construction," *IEEE Trans. on Commun.*, vol. 58, no. 5, pp. 1383-1396, May 2010.

[12] D. Divsalar and L. Dolecek, "Enumerators for protograph-based ensembles of nonbinary LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint-Petersburg, Russia, Jul.- Aug. 2011, pp. 913 - 917.

[13] I. Andriyanova, D. Maurice, and J-P. Tillich, "Quantum LDPC codes obtained by non-binary constructions," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Boston, MA, Jul. 2012, pp. 343 - 347.

[14] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 633-643, Apr. 2007.

[15] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. on Commun.*, vol. 58, no. 5, pp. 1365-1375, May 2010.

[16] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic,, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. on Inf. Theory*, vol. 56, no. 1, pp. 181-201, Jan. 2010.

[17] O. Milenkovic, E. Soljanin, and P. Whiting, "Asymptotic spectra of trapping sets in regular and irregular LDPC code ensembles," *IEEE Trans. on Inf. Theory*, vol. 53, no. 1, pp. 39-55, Jan. 2007.

[18] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2, d_c)-$LDPC codes over GF(q) using their binary images," *IEEE Trans. on Commun.*, vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

[19] V. Rathi and I. Andriyanova, "Some results on MAP decoding of non-binary LDPC codes over the BEC,"' *IEEE Trans. on Inf. Theory*, vol. 57, no. 4, pp. 2225-2242, Apr. 2011.

[20] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, pp. 533 - 547, Sep. 1981.

[21] J. Pearl, "Reverend Bayes on inference engines: a distributed hierarchical approach," in *Proceedings of American Association for Artificial Intelligence National Conference on AI*, pp. 133 - 136, 1982.

[22] T. J. Richardson, and R. L. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.

[23] T. J. Richardson, "Error floors of LDPC codes," in *Proc. 41st Allerton Conf. on Comm., Cont., and Comp.*, Allerton, IL, Oct. 2003, pp. 1426-1435.

[24] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analysis of stopping constellation distribution for irregular non-binary LDPC code ensemble," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint-Petersburg, Russia, Jul.- Aug. 2011, pp. 1101-1105.

[25] T. Nozaki, K. Kasai, and K. Sakaniwa, "'Error floors of non-binary LDPC codes" in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 729 - 733.

[26] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analysis of error floors of generalized non-binary LDPC codes over q-ary memoryless symmetric channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Boston, MA, Jul. 2012, pp. 2341 - 2345.

[27] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, "Non-binary protograph-based LDPC codes: enumerators, analysis, and designs," Submitted to *IEEE Trans. on Inf. Theory*, 2013.

[28] A. Poloni, S. Valle, and S. Vincenti, "NB-LDPC: Absorbing set and importance sampling," *Inter. Symp. on Turbo Codes and Iter. Inf. Proc. (ISTC)*, Gothenburg, Sweden, Aug. 2012, pp. 101-105.

[29] B. Amiri, C. W. Lin and L. Dolecek, "Asymptotic distribution of absorbing sets and fully absorbing sets for regular sparse code ensembles," *IEEE Trans. on Commun.*, vol. 61, no. 2, pp. 455 - 464, Feb. 2013.

[30] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," IPN Progress Report, Tech. Rep. 42-154, Aug. 2003.

[31] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd Ed. Wiley-Interscience, 2006.

[32] R. G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA, MIT Press, 1963.

[33] S. Abu-Surra, D. Divsalar and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Trans. on Inf. Theory*, vol. 57, no. 2, pp. 858-886, Feb. 2011.

[34] D. Divsalar and L. Dolecek, "On the typical minimum distance of protograph-based non-binary LDPC codes," in *UCSD Workshop on Inf. Theory and Its Applications (ITA)*, San Diego, CA, Feb, 2012.

[35] M. M. Mansour and N.R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. on VLSI Systems*, vol. 11, no. 6, pp. 976 - 996, Dec. 2003.

[36] L. Dolecek, "On absorbing sets of structured sparse graph codes," in *Proc. Info. Theory and Applications (ITA) Workshop*, San Diego, CA, Feb. 2010.

[37] S. Zhang and C. Schlegel, "Causes and dynamics of LDPC error floors on AWGN channels," in *Proc. 49th Allerton Conf.*, pp.1025-1032, Sep. 2011.

[38] J. Kang, Q. Huang, S. Lin and K. Abdel-Ghaffar, "An iterative decoding algorithm with backtracking to Lower the error-floors of LDPC codes," *IEEE Trans. on Comm.*, vol. 59, no. 1, pp. 64-73, Jan. 2011.

[39] J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181-2191, 1999.

[40] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, pp. 5274-5289, Oct. 2010.

[41] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803-834, 2011.

[42] M. Karimi and A. H. Banihashemi, "An efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 58, pp. 6942 - 6958, Nov. 2012.

[43] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. on Turbo Codes*, Brest, France, Sep. 2000.

[44] A. Bazarsky, N. Presman, and S. Litsyn, "Design of non-binary quasi-cyclic LDPC codes by ACE optimization" Submitted to *IEEE Trans. on Inf. Theory*, 2013, Available at: http://arxiv.org/abs/1304.7487.

[45] L. Zeng *et al.*, "Constructions of nonbinary quasi-cyclic LDPC codes: a finite field approach," *IEEE Trans. Commun.*, Apr. 2008.

[46] J. Huang *et al.*, "Large-girth nonbinary QC-codes of various lengths," *IEEE Trans. Commun.*, Dec. 2010.

[47] L. Zhang *et al.*, "Quasi-cyclic LDPC codes: An algebraic construction, rank analysis, and codes on Latin squares," *IEEE Trans. Commun.*, 2010.

[48] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analysis of error floors of non-binary LDPC codes over MBIOS channel," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1 - 5.

[49] B. Zhou, J. Kang, Y. Y. Tai, S. Lin, and Z. Ding, "High performance non-binary quasi-cyclic LDPC codes on Euclidean geometries," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1298 - 1311, 2009.

[50] B.-Y. Chang, D. Divsalar, and L. Dolecek, "Non-binary protograph-based LDPC codes for short blocklengths," in *IEEE Inf. Theory Workshop (ITW)*, Lausanne, Switzerland, Sept. 2012, pp. 282 - 286.

[51] X.Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386 - 398, Jan. 2005.

[52] J. Huang, S. Zhou, and P. Willett, "Nonbinary LDPC coding for multicarrier underwater acoustic communication," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1684 - 1696, Dec. 2008.

[53] R.-H. Peng and R. R. Chen, "Design of nonbinary quasi-cyclic LDPC cycle codes," in *Proc. IEEE Inform. Theory Workshop*, Tahoe City, CA Sep. 2007, pp. 13-18.

[54] C-C. Wang *et al.*, "Finding all small error-prone substructures in LDPC codes," *IEEE Trans. Inform. Theory*, Apr. 2009.

[55] E. A. Bender, "The asymptotic number of nonnegative integer matrices with given row and column sums," *Discrete Math*, vol. 10, no. 2, pp. 217-223, 1974.

[56] S. Litsyn and V. Shevelev, "On ensembles of low-density parity-check codes: asymptotic distance distributions," *IEEE Trans. on Inform. Theory*, vol. 48, no. 4, pp. 887-908, Apr. 2002.

[57] S. Abu-Surra, D. Divsalar, and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Trans. on Inform. Theory*, vol. 57, no. 2, pp. 858-886, Feb. 2011.

[58] J. Kliewer, K. S. Zigangirov, C. Koller, and D. J. Costello, Jr., "Coding theorems for repeat multiple accumulate codes," submitted to *IEEE Trans. on Info. Theory*, Oct. 2008. Available at http://arxiv.org/abs/0810.3422/.

[59] H. D. Pfister, *On the Capacity of Finite State Channels and the Analysis of Convolutional Accumulate-m Codes*, Ph.D. Thesis, University of California, San Diego, 2003.

[60] C. Koller, A. Graell i Amat, J. Kliewer, and D. J. Costello, "Trapping set enumerators for repeat multiple accumulate code ensembles," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Seoul, Korea, Jun. - Jul. 2009, pp. 1818-1823.

[61] M. El-Khamy, "New approaches to the analysis and design of Reed-Solomon related codes," Ph.D. Thesis, California Institute of Technology, Sep. 2006.

[62] S. Benedetto, D. Divsalar, G.Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. on Inf. Theory*, vol. 44, no. 3, pp. 909 - 926, May 1998.

[63] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, Jr., "Absorbing set characterization of array-based spatially-coupled LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, Jun. 2014, pp. 886 - 890.

[64] B. Amiri, A. Resisizadeh, J. Kliewer, and L. Dolecek, "Optimized design of finite-length spatially-coupled codes: an absorbing set-based analysis," *Technical Report*, [online]. Available: http://amiri.bol.ucla.edu/TechnicalReport.pdf

[65] I. Andriyanova and A. Graell i Amat, "Threshold saturation for non-binary SC-LDPC codes on the binary erasure channel," *Submitted to IEEE Trans. on Inf. Theory*, 2013. [online]. Available: http://arxiv.org/abs/1311.2003

[66] L. Wei, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello, Jr., "Design of spatially-coupled LDPC codes over GF(q) for windowed decoding," *Submitted to IEEE Trans. on Inf. Theory*, 2014. [online]. Available: http://arxiv.org/abs/1411.4373

[67] K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, Jr., "Performance comparison of LDPC block and spatially-coupled codes over GF(q)," Submitted to *IEEE Trans. Commun.*, 2014.