Title
Absorbing Set Distributions, Quantization and Practical Message Passing Decoders

Permalink
https://escholarship.org/uc/item/2q75b1xc

Author
Amiri, Behzad

Publication Date
2012

Peer reviewed|Thesis/dissertation
Absorbing Set Distributions, Quantization and Practical Message Passing Decoders

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical Engineering

by

Behzad Amiri

2012
© copyright by
Behzad Amiri
2012
ABSTRACT OF THE THESIS

Absorbing Set Distributions, Quantization and Practical Message Passing Decoders

by

Behzad Amiri

Master of Science in Electrical Engineering

University of California, Los Angeles, 2012

Professor Lara Dolecek, Chair

It is well recognized that low-density parity-check (LDPC) codes can suffer from an error floor when decoded iteratively. This performance degradation is often attributed to the class of objects known as trapping sets. As a subset of the trapping set collection, there exists a class of graphical structures called the absorbing sets. An absorbing set is a combinatorially-defined object; in particular a fully absorbing set is stable under bit-flipping decoding. By construction, there can exist trapping sets that are not stable under such a decoder. As a result, for finite-precision, iterative decoding algorithms used over additive channels, absorbing sets can describe decoding errors more accurately than the broader class of trapping sets. In the first part of this thesis, we compute the normalized logarithmic asymptotic distributions of absorbing sets and fully absorbing sets, including elementary (fully) absorbing sets. 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.

To reduce implementation complexity, the messages in a practical message passing decoder are necessarily quantized. Absorbing regions act as “decoding regions” around absorbing sets. In the second part of this thesis, we take a closer look at the interplay between quantization and absorbing regions. We provide a study of a range of quantization choices, the impact of quantization on the candidate absorbing regions, and derive guidelines for practical decoders. We show that, due to the non-linear dynamics of message passing decoders, coarser quantization may in fact perform better than finer quantization. Results of this type of work can be particularly useful in designing high performance decoders for very high-reliability storage systems, such as emerging data storage hard disk and solid state drives.
The thesis of Behzad Amiri is approved.

Alan J. Laub
John Villasenor
Lara Dolecek, Committee Chair

University of California, Los Angeles
2012
DEDICATION

To my parents.
Contents

1 Introduction 3
  1.1 Background .................................................. 3
  1.2 Graphical Modeling of Linear Codes ......................... 4
  1.3 LDPC codes .................................................. 7
    1.3.1 Array-based LDPC codes ................................ 9
  1.4 Error floor in LDPC codes .................................. 10
    1.4.1 Trapping set ............................................. 11
    1.4.2 Absorbing set ............................................ 11
  1.5 Thesis Overview ............................................. 13

2 Asymptotic Distribution of Absorbing Sets Regular Code Ensembles 15
  2.1 Introduction ................................................ 15
    2.1.1 Random Matrix Enumeration ............................ 17
  2.2 Asymptotic Distribution of Absorbing Sets and Fully Absorbing Sets
    for Regular Code Ensembles .................................. 19
  2.3 Numerical Results .......................................... 29
    2.3.1 Implications for certain structured LDPC ensembles .... 32

3 Effect of quantization parameters on the performance of BP decoder 35
  3.1 Introduction ................................................. 35
  3.2 BP Decoding ................................................ 36
  3.3 Quantization Parameters .................................... 37
  3.4 Experimental results ....................................... 38
Chapter 1

Introduction

1.1 Background

In 1948, the concept of reliable data transmission over noisy channels was introduced by Shannon [1]. He proposed a system consisting of an encoder, a channel and a decoder. Shannon revealed that beyond a 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, by which the decoding error approaches zero exponentially fast by increasing the block-length of the code. However, the theorem did not present any practical method for designing channel codes. Furthermore, the random coding scheme suggested in the proof is a computationally intensive algorithm both in the encoding and decoding parts. After Shannon’s paper, there has been a huge amount of research on finding codes which are practically easy to encode and decode and that could also approach the channel capacity.

Linear codes are channel codes whose codewords create 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 for low complexity encoding. Moreover, many algebraic linear codes have polynomial decoding time, such as Hamming [3], BCH [4], Reed-Solomon [5], and convolutional codes [6]. However, none of these codes could reach the capacity of Additive White Gaussian Noise (AWGN) channels. The discovery of capacity-approaching turbo codes [7] and rediscovery of low-density parity-check (LDPC) codes [8], [9] started a new coding era centered on graph-based constructions.

Excellent performance of LDPC codes has resulted in their growing use in many applications. These codes reach outstanding near-capacity performance with an acceptable encoding/decoding complexity [10]-[12]. Analysis and design of LDPC codes and their decoders have attracted a lot of recent attention.

1.2 Graphical Modeling of Linear Codes

Channel codes whose codewords create a linear vector space over a field \( F \) are called linear codes. In particular, a code \( C \) over the field \( F \) is called linear if

\[
\forall x_1, x_2 \in C, \forall \alpha_1, \alpha_2 \in F : \alpha_1 x_1 + \alpha_2 x_2 \in C.
\]  

(1.1)

Assume that \( C \) has a block-length of \( n \). Therefore, codewords of \( C \) are members of \( F^n \). There is an integer \( k \), called the dimension of the code, such that

\[
|C| = |F|^k.
\]  

(1.2)

For any linear code, there exists a matrix called the parity-check matrix such that its codewords are in the null space of the code of the code. Thus, linear codes can be defined based on their parity-check matrices as well.

In other words, we also define the linear code \( C \) as the set of codewords \( x = (x_1, x_2, ..., x_n) \) over \( F^n \) which satisfy the parity-check equation
where $H$ is an $(n - k) \times n$ parity-check matrix with entries from $F$, and 0 is the $(n - k) \times 1$ all-zero vector.

In 1981, Tanner proposed a graphical representation of a parity-check matrix of a linear code [13]. Pearl invented a belief propagation (BP) algorithm as a message passing algorithm to perform inference over Bayesian networks [14]. Now, BP is used in many signal processing, digital communication, and artificial intelligence algorithms such as Viterbi algorithm, turbo decoding, Kalman filtering [15], 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 [16] 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 [17]. Consequently, the Tanner graph representation of a code has an important role in the decoding of an LDPC code.

**Definition 1.** The *Tanner graph* of a code with the $(n - k) \times n$ parity-check matrix $H$ is a bipartite graph consists of $2n - k$ vertices 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 $i$th row and $j$th 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 1.1 shows the Tanner graph of a length-7, Hamming code associated with the parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (1.4)$$
Figure 1.1: Tanner graph associated with binary parity check matrix in equation 1.4.

Figure 1.2 represents the Tanner graph of a length-5 LDPC code over $GF(4)$, with the parity-check matrix

$$H = \begin{bmatrix}
\alpha & \alpha^2 & \alpha^3 & 0 & 0 \\
0 & \alpha & \alpha^2 & \alpha^3 & 0 \\
0 & 0 & \alpha & \alpha^2 & \alpha^3
\end{bmatrix}, \quad (1.5)$$

Figure 1.2: Tanner graph associated with non-binary parity check matrix in equation 1.5.

where $\alpha$ 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. 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.

### 1.3 LDPC codes

Gallager first introduced the LDPC codes in 1963, in his PhD thesis [8]. However, LDPC codes were forgotten for about thirty years, as a result of their high computational and storage requirements. Zyablov and Pinsker discovered a decoding algorithm for codes with sparse parity-check matrices [18]. The construction of LDPC codes was generalized by Tanner and he introduced bipartite graph for graphical representation of these codes [13]. Afterwards, the relations between different message passing algorithms was established by Wiberg [19]. MacKay showed that LDPC codes can accomplish even better performance than the best turbo codes [20].

Investigation of LDPC codes over erasure channels [21] inspired the design of capacity-approaching LDPC codes over erasure channels [22]. Later, the design of codes with performance close to Shannon’s limit were generalized to AWGN channels [23] and other channels as well, e.g., [24].

LDPC codes are linear codes that have a sparse parity-check matrix. These codes, like any linear code, can be denoted by their corresponding Tanner graphs. In the Tanner graph, the \( n \) columns of the parity-check matrix correspond to the bit nodes and the \( n-k \) rows of the parity-check matrix to the check nodes. There is an edge between variable node \( j \) and check node \( i \) if \( H_{i,j} \neq 0 \) and this edge is related with the constant \( H_{i,j} \). In the case of binary LDPC codes, that is LDPC codes over \( GF(2) \), all nonzero elements of parity-check matrix are 1, so the edge does not have weights. we define the number of edges connected to each node in the Tanner graph as the degree of the node.
Definition 2. A \((d_v, d_c)\)-regular LDPC code is an LDPC code such that each check node has degree \(d_c\) and each bit node has degree \(d_v\). Similarly, we say that the parity-check matrix of a regular LDPC code has all column-weights equal to \(d_v\) and all row-weights equal to \(d_c\).

Generally, the degrees of the nodes can be different. Irregular LDPC codes are codes whose variable nodes, and/or check nodes, have different degrees. An LDPC code is often defined by following two degree distributions:

\[
\lambda(x) = \sum_{i=1}^{l_{\text{max}}} \lambda_i x^{i-1},
\]

and

\[
\rho(x) = \sum_{i=1}^{r_{\text{max}}} \rho_i x^{i-1},
\]

where \(\lambda_i (\rho_i)\) is the fraction of edges connected to degree \(i\) bit nodes (check nodes) and \(l_{\text{max}} (r_{\text{max}})\) is the maximum degree of bit nodes (check nodes). Similar to the degree distributions defined for edges, we can define node perspective degree distributions for both bit and check nodes as

\[
\Lambda_i x^i = \frac{\int_0^x \lambda(u) du}{\int_0^1 \lambda(u) du},
\]

and

\[
P_i x^i = \frac{\int_0^x \rho(u) du}{\int_0^1 \rho(u) du},
\]

where \(\Lambda_i\) for \(1 < i < l_{\text{max}}\) represents the fraction of degree-\(i\) variable nodes relative to the total number of variable nodes in the Tanner graph. \(P_i\) for \(1 < i < r_{\text{max}}\) represents the fraction of degree-\(i\) check nodes relative to the total number of check nodes in the Tanner graph.

Given the degree distributions of a code, one can generate a random code corresponding to the distributions. First, we define \(n\) bit and \(n - k\) check nodes. The number
of edges connected to each bit node is assigned such that a fraction \( \lambda_i \) of bit nodes have degree equal to \( i = 1, 2, \ldots, l_{\text{max}} \). Likewise, degrees of check nodes are assigned such that a fraction \( \rho_i \) of the edges is connected to degree-\( i \) check nodes for \( i = 1, 2, \ldots, r_{\text{max}} \). Then, using a random permutation, we connect the corresponding edges to bit node and check nodes.

### 1.3.1 Array-based LDPC codes

Array-based codes are a type of structured regular LDPC codes parameterized by a pair of integers \((p, \gamma)\), such that \( \gamma \leq p \), where \( p \) is a prime number. We form \( p\gamma \times p^2 \) parity check matrix \( H_{p,\gamma} \), \[ H_{p,\gamma} = \begin{bmatrix} I & I & I & \cdots & I \\ I & \sigma & \sigma^2 & \cdots & \sigma^{(p-1)} \\ I & \sigma^2 & \sigma^4 & \cdots & \sigma^2(p-1) \\ \hdotsfor{6} \\ I & \sigma^{(\gamma-1)} & \sigma^{2(\gamma-1)} & \cdots & \sigma^{(p-1)(\gamma-1)} \end{bmatrix} , \] where \( \sigma \) is a \( p \times p \) circulant matrix such that

\[
\sigma = \begin{bmatrix}
0 & 0 & \ldots & 0 & 1 \\
1 & 0 & \ldots & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 \\
\hdotsfor{5} \\
0 & 0 & \ldots & 1 & 0
\end{bmatrix} .
\]

The rate of this code is \( R = 1 - \frac{\gamma p - \gamma + 1}{p^2} \). Fan [25] proved that array-based LDPC codes have very good performance in the low SNR region, and they have then been used for a number of applications, including digital subscriber lines [26] and magnetic recording [27].
We will study the effect of changing the quantization parameters in decoding of array-based codes.

1.4 Error floor in LDPC codes

LDPC codes are famous for their outstanding performance at moderate signal-to-noise ratios (SNRs). However, for adequately high SNRs, some of these codes exhibit a rapid change in the slope of the curve representing the bit-error rate (BER) vs SNR. This variation in the slope of the BER curve is known as the error-floor of the code. For a large class of codes, including array-based LDPC codes, error-floors appear at fairly high BERs. This phenomenon impedes the use of such codes in storage and deep-space communication systems that need tremendously low BERs. For many codes, error-floors cannot be observed directly, because they are out of reach of standard simulation techniques.

Many works have been done to characterize the cause of the error floor in LDPC codes. It is shown that error floors are associated with the suboptimality of practical message-passing decoders on graphs with cycles. Mackay and Postol [28] recognized that some classes of non-codewords, which they termed near-codewords, can result in a failure in the decoder; particularly an \((a, b)\) near-codeword is a binary string of weight \(a\) whose syndrome has weight \(b\) (for the received string \(x\), the syndrome \(s\) is defined as \(s = Hx^t\)). They showed that in the simulation of a rate \(1/2\) LDPC code with block-length 2640 based on the Margulis construction, \((12, 4)\) and \((14, 4)\) near-codewords are the key reasons for the error floor of this code when used for the transmission over an additive white Gaussian noise (AWGN) channel. They also claimed that the minimum distance of the code is considerably more than the weight \(a\) of the observed near-codewords. In [29], authors defined a similar concept called stopping sets, which are the main performance limitation under iterative decoding for LDPC codes over the binary erasure channel (BEC). Since stopping sets have a combinatorial characterization, their distribution within ensembles of graphs can be analyzed, though with some
difficulty. As a result, in [30], authors have established the stopping set enumerator for different ensembles of LDPC codes.

Although stopping sets are useful in characterizing the errors in error floor region for BEC channels, they cannot be used to determine the performance of LDPC decoder in other types of channels. Therefore, trapping sets were introduced in [31] to capture decoding errors under iterative decoding algorithms for various types of channels.

1.4.1 Trapping set

The definition of trapping sets was presented by Richardson in [31] to explain the errors of iterative deciders for channel other than BEC channel.

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 definition is from [32].

**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 [32] 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.

1.4.2 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 [33] are combinatorial objects in Tanner graph that are guaranteed to be stable under a bit-flipping decoder.

**Definition 4.** An $A_{a,b}$ 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 $C \setminus O(D)$.

Fully absorbing sets [33] are the subclass of absorbing sets wherein all bit nodes have strictly fewer unsatisfied than satisfied checks.

**Definition 5.** An $A^{(f)}_{a,b}$ fully absorbing set of size $(a,b)$ is an absorbing set of size $(a,b)$ such that in the subset of bit nodes $V \setminus D$, each element has strictly fewer neighbors in $O(D)$ than in $C \setminus O(D)$.

Clearly, there can exist even small trapping sets that do not fulfill the combinatorial requirements of fully 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 6.** An elementary (fully) 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.

Fig. 1.3 shows an example of a $(4,2)$ fully absorbing set in the Tanner graph of an LDPC code with 25 bit nodes each of degree 3, and with 15 check nodes each of degree 5. Since the absorbing sets are a subclass of trapping sets, the configuration in Fig. 1.3 is also a trapping set.

On the other hand, trapping sets do not always satisfy absorbing set constraints. Fig. 1.4 shows an example of a $(4,6)$ trapping set, which is not an absorbing set.
Figure 1.3: An example of a (4, 2) fully absorbing set. The circles form the set $V$, and 4 black circles form the fully absorbing set $D$. The squares form the set $C$. The 5 gray squares constitute the set of checks that are satisfied with respect to $D$, and the 2 black squares constitute the set $O(D)$ (checks that are not satisfied with respect to $D$). Note that all bit nodes have more connections to $C \setminus O(D)$ than to $O(D)$.

Figure 1.4: This figure shows an example of a (4, 6) trapping set, which is not an absorbing set, on the same Tanner graph as in Fig. 1.3. Again, black circles represent the set $D$ and black squares represent the set $O(D)$. Note that the leftmost black circle is connected to 2 checks in $O(D)$ and 1 check in $C \setminus O(D)$.

1.5 Thesis Overview

In chapter 2, we first introduce an approach for enumeration of random matrices by dividing them into two sub-matrices and enumeration of those sub-matrices. Then, this approach is used to find the asymptotic distribution of absorbing sets and fully absorbing sets for regular LDPC code ensembles. Following this analytical presentation, numerical result are provided. Chapter 3 analyzes the effect of quantization parameters of BP decoder performance. As a result of the intricate nature of message passing decoders, it turns out that coarser quantization may in fact perform better than finer quantization. Experimental results are demonstrating this concept. Finally, chapter 4
delivers the conclusion of this work.
Chapter 2

Asymptotic Distribution of Absorbing Sets Regular Code Ensembles

2.1 Introduction

It is well known that low-density parity check (LDPC) codes [8] can offer excellent performance even when decoded using suboptimal but low-complexity message passing algorithms. It is also well recognized that in the region of high signal-to-noise ratio (SNR), the performance of LDPC codes is dominated by certain non-codeword objects vying with codewords to be the outputs of the decoder.

Early work [28] already explained the resulting performance degradation in terms of near codewords. Concurrently, trapping sets were introduced in [31] to capture decoding errors under iterative decoding algorithms for various types of channels. The relationship between performance and trapping sets for practical codes was explored in [31] and [34]. An analytical approach based on enumerating parity checks of LDPC codes was developed in [32] to study trapping set distributions in the asymptotic regime (i.e., in the limit of large block lengths). Methods for enumerating and identifying trapping sets of certain finite-length LDPC codes were presented in [35] and [36]. Asymptotics of the trapping sets of structured LDPC code ensembles were studied in [37] for protograph-based LDPC codes, in [38] for repeat accumulate accumulate LDPC codes,
and in [39] for LDPC convolutional codes.

Stopping sets were introduced to combinatorially capture decoding errors over a binary erasure channel (BEC) [40]. Distributions of stopping sets within LDPC ensembles were characterized in [41] and [42]. As a combinatorial counterpart of stopping sets, but more suitable for channels with additive noise, absorbing sets were used in [33] to characterize the error floors of a representative class of LDPC codes. Absorbing sets are a combinatorially defined subclass of trapping sets, and in particular fully absorbing sets are stable under a bit-flipping decoder. In the finite-length regime, recent work has investigated methods to efficiently search for dominant absorbing sets [43], [44], to analyze absorbing sets for representative codes [33], [45], and to design codes free of detrimental absorbing/trapping sets [46] and [47].

Since not all trapping sets are problematic in practical (finite-precision) iterative decoding algorithms, it is worthwhile to characterize the subclass of trapping sets that is the main cause of errors under such decoders. The first step towards an ambitious goal of performance characterization of entire LDPC ensembles under a broad class of iterative decoders is to focus on the bit-flipping-like decoding algorithms. These popular decoding algorithms serve as useful proxies for more sophisticated decoders [48]. It is therefore of interest to inspect absorbing sets and fully absorbing sets, and to investigate when and how their distributions differ from the trapping set distributions ensemble-wide.

The contribution of this chapter is to establish the asymptotic analysis of (fully) absorbing sets for regular LDPC code ensembles, and to quantify the difference between the asymptotic distributions of absorbing sets and trapping sets. When the normalized logarithmic asymptotic absorbing set distribution is almost identical to the normalized logarithmic asymptotic trapping set distribution, most of these trapping sets are in fact absorbing sets, and these trapping sets can represent the decoding errors more 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 to
efficiently search for absorbing sets (or relevant trapping sets), [43]. The presented results are based on the computational approach first introduced in [32] for the trapping set analysis.

In Section 2.1.1, we will state the key computational method which is used in the later sections. In Section 2.2, we present an analytical approach for the asymptotic enumeration of absorbing sets and fully absorbing sets in regular LDPC code ensembles. We derive simplified expressions for the normalized logarithmic asymptotic distribution of (fully) absorbing sets when all bit nodes have degree 3, as well as when all bit degrees are 4. In Section 2.3, we provide numerical results for representative bit and check node degrees. We also compare the distributions of absorbing sets and fully absorbing sets with that of trapping sets for regular LDPC code ensembles and for certain structured LDPC codes (including repeat accumulate codes and protograph-based codes).

2.1.1 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 7.** Let $\Lambda_{l,r}^{n,m}$ be the set of all $n \times m$ binary matrices with row-weight vector $l = (l_1, l_2, ..., l_n)$ and column-weight vector $r = (r_1, r_2, ..., r_m)$, 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 $\Lambda_{l,r}^{n,m}$.

We use the following theorem from [49] 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 [32].

**Theorem 1.** (cf. [49]) For given $l = (l_1, l_2, ..., l_n)$ and $r = (r_1, r_2, ..., r_m)$, let $t_{n,m}(l,r) = |\Lambda_{l,r}^{n,m}|$ be the cardinality of the collection $\Lambda_{l,r}^{n,m}$, 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 \( n \to \infty \),

\[
t_{n,m}(l, r) = \frac{f!}{\prod_{i=1}^{n} l_i! \prod_{j=1}^{m} r_j!} \cdot \exp \left[ -\frac{1}{2f^2} \sum_{i=1}^{n} l_i (l_i - 1) \sum_{j=1}^{m} r_j (r_j - 1) \right], \tag{2.1}
\]

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

Lemma 1 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 \( l, r, m, n \), and \( n_u < n \), and an integer-valued vector \( r_u = (r_1, \ldots, r_m) \) with each entry \( r_j \) satisfying \( 0 \leq r_j \leq r \), we let

- \( \Lambda_{n,m}^{l,r} \) be the set of \( n \times m \) binary matrices with rows weighted \( l \) and columns weighted \( r \) (cf. Definition 7),

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

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

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

**Lemma 1.** Let \( \Lambda_{n,m}^{l,r}, \Lambda_{n_u,m}^{l,r_u} \) and \( \Lambda_{n_d,m}^{l,r_d} \) be the three sets of matrices as defined above. The cardinality of the set \( \Lambda_{n,m}^{l,r} \) is expressed in terms of cardinalities of constituent sets as,

\[
\left| \Lambda_{n,m}^{l,r} \right| = \sum_{\{r_u: 1 \leq j \leq m, 0 \leq r_j \leq r\}} \left| \Lambda_{n_u,m}^{l,r_u} \right| \cdot \left| \Lambda_{n_d,m}^{l,r_d} \right|. \tag{2.2}
\]

**Proof.** For a particular choice of the vector \( r_u \), let \( M_u \in \Lambda_{n_u,m}^{l,r_u} \) and \( M_d \in \Lambda_{n_d,m}^{l,r_d} \). By pairing up \( M_u \) and \( M_d \), such that \( M = \left[ \begin{array}{c} M_u \\ M_d \end{array} \right] \), we obtain a regular matrix with rows weighted \( l \) and columns weighted \( r \). There are \( \left| \Lambda_{n_u,m}^{l,r_u} \right| \cdot \left| \Lambda_{n_d,m}^{l,r_d} \right| \) different regular matrices for a particular vector \( r_u \). Summing over all possible choices of \( r_u \) leads the overall count. \( \square \)
For convenience, we let $G_{n,m}^{l,r}$ be the set of all Tanner graphs corresponding to the parity check matrices whose transposes are in the $\Lambda_{n,m}^{l,r}$ collection.

## 2.2 Asymptotic Distribution of Absorbing Sets and Fully Absorbing Sets for Regular Code Ensembles

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

$$ e^{l,r}(\theta, \lambda) \triangleq \lim_{n \to \infty} \frac{1}{n} \log p_{a,b,n}^{l,r} = \lim_{n \to \infty} \frac{1}{n} \log \frac{z_{a,b,n}^{l,r}}{\left| \Lambda_{n,m}^{l,r} \right|}, \quad (2.3) $$

where $\theta = \frac{a}{n}$, $\lambda = \frac{b}{n}$ and $p_{a,b,n}^{l,r}$ is the average number of size $(a, b)$ absorbing sets in a Tanner graph in $G_{n,m}^{l,r}$. Here, $z_{a,b,n}^{l,r}$ is the number of $(a, b)$ absorbing sets over all Tanner graphs in $G_{n,m}^{l,r}$. 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

$$ e^{(f)l,r}(\theta, \lambda) \triangleq \lim_{n \to \infty} \frac{1}{n} \log p_{a,b,n}^{(f)l,r} = \lim_{n \to \infty} \frac{1}{n} \log \frac{z_{a,b,n}^{(f)l,r}}{\left| \Lambda_{n,m}^{l,r} \right|}, \quad (2.4) $$

where $p_{a,b,n}^{(f)l,r}$ is the average number and $z_{a,b,n}^{(f)l,r}$ is the total number of $(a, b)$ fully absorbing sets in $G_{n,m}^{l,r}$.

The normalized logarithmic asymptotic distributions of elementary (fully) absorbing sets are defined analogously to (2.3) and (2.4); 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 $G_{n,m}^{l,r}$. 
For $1 \leq i \leq a$, let $\sigma_i$ denote the number of edges that connect the $i^{th}$ bit node in $A_{a,b}$ to satisfied check nodes, and for $1 \leq j \leq m$, let $\delta_j$ denote the number of edges that connect the $j^{th}$ check node to the bit nodes in $A_{a,b}$. Theorem 2 establishes the asymptotic logarithmic scaling of the average number of $(a,b)$ absorbing sets in $G_{n,m}^{lr}$.

Here, "$p_{a,b,n}^{lr} \approx w_{a,b,n}^{lr}$" means "$\lim_{n \to \infty} \frac{1}{n} \log p_{a,b,n}^{lr} = \lim_{n \to \infty} \frac{1}{n} \log w_{a,b,n}^{lr}$".

**Theorem 2.** Let $0 < \zeta, \theta, \lambda < 1$ where $\zeta = \frac{m}{n} = \frac{l}{r}$. Then,

$$p_{a,b,n}^{lr} \approx \sum_{\{\delta_j, \sigma_i : 1 \leq j \leq m, 1 \leq i \leq n, S\}} \left( \frac{n}{a} \right) \left( \frac{m}{b} \right) \prod_{i=1}^{a} \left( \frac{l}{\sigma_i} \right) \prod_{j=1}^{m} \left( \frac{r}{\delta_j} \right),$$

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

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

**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 $H$):

$$H^T = \begin{bmatrix}
M_1 & M_2 \\
M_3
\end{bmatrix},$$

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.
Let us define the following collections of matrices:

\[ |\Lambda_1| \equiv \left( \frac{m-b}{\prod_{i=1}^{a} \sigma_i! \prod_{j=1}^{m-b} \delta_j!} \right) \cdot \exp \left[ - \sum_{i=1}^{a} \frac{\sigma_i(\sigma_i - 1)}{2} \left( \sum_{j=1}^{m-b} \delta_j \right)^2 \right], \tag{2.7} \]

\[ |\Lambda_2| \equiv \left( \frac{\sum_{j=m-b+1}^{m} \delta_j!}{\prod_{i=1}^{a} (l - \sigma_i)! \prod_{j=m-b+1}^{m} \delta_j!} \right) \cdot \exp \left[ - \sum_{i=1}^{a} \frac{(l - \sigma_i)(l - \sigma_i - 1)}{2} \left( \sum_{j=m-b+1}^{m} \delta_j \right)^2 \right], \tag{2.8} \]

Then, for \(1 \leq j \leq m\), \(\delta_j\) is the weight of the \(j^{th}\) column of the submatrix \([M_1|M_2]\), and for \(1 \leq i \leq a\), \(\sigma_i\) 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 \(\sigma_i > \frac{1}{2}\) 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:

\[ \Lambda_1 = \{ \forall M \in \Lambda_{a,m-b} : \text{ for } 1 \leq i \leq a, \sum_{h=1}^{m-b} M(i, h) = \sigma_i; \text{ for } 1 \leq j \leq m-b, \sum_{g=1}^{a} M(g, j) = \delta_j \} \],

\[ \Lambda_2 = \{ \forall M \in \Lambda_{a,b} : \text{ for } 1 \leq i \leq a, \sum_{h=1}^{b} M(i, h) = l - \sigma_i; \text{ for } 1 \leq j \leq b, \sum_{g=1}^{a} M(g, j) = \delta_{m-b+j} \} \],

\[ \Lambda_3 = \{ \forall M \in \Lambda_{n-a,m} : \text{ for } 1 \leq i \leq n-a, \sum_{h=1}^{m} M(i, h) = l; \text{ for } 1 \leq j \leq m, \sum_{g=1}^{n-a} M(g, j) = r - \delta_j \} \].

The asymptotic cardinalities of sets \(\Lambda_1, \Lambda_2, \Lambda_3\) and \(\Lambda_{n,m}^{(L)}\) are computed using Theorem 1, and are shown in (2.7) through (2.10).
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 1 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:

$$
\sum_{\{\delta_j, \sigma_i: 1 \leq j \leq m, 1 \leq i \leq n, S\}} \binom{n}{a} \binom{m}{b} |\Lambda_1||\Lambda_2||\Lambda_3|, 
$$

(2.11)

where the condition set $S$ is given by (2.6). Therefore, the average number of size $(a, b)$ absorbing sets in a Tanner graph in $G_{n,m}^{l,r}$ is

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

(2.12)

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 [32], [37]. It is therefore of interest to also quantify distributions of elementary absorbing sets in this regime.

**Corollary 1.** The normalized logarithmic asymptotic distribution of $(a = \theta n, b = \lambda n)$ elementary absorbing sets in $G_{n,m}^{3r}$ is given by (recall that “E” stands for “elemen-
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, \( \delta_j \)'s have values 0, 1 or 2. For \( m - b + 1 \leq j \leq m \), \( \delta_j = 1 \) since each unsatisfied check has only one edge connected to the absorbing set.

Following the constraints on values of \( \sigma_i \)'s (for \( 1 \leq i \leq a, \frac{l}{2} < \sigma_i \leq l \)), \( \sigma_i \)'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 \( \lambda n \). Each row of \( M_2 \) is weighted 1 or 0, and \( \lambda n \) number of rows of \( M_2 \) are weighted 1. Thus, \( \lambda n \) number of \( \sigma_i \)'s are equal to 2 and the rest of \( \sigma_i \)'s are equal to 3. Since the total weight in \( [M_1|M_2] \) is \( 3\theta n \), it follows that \( \frac{(3\theta - \lambda)\lambda n}{2} \) number of \( \delta_j \)'s are equal to 2 and rest of them are equal to 0. By substituting in the numerical values for \( \delta_j \)'s and \( \sigma_j \)'s in (2.5), we have:

\[
e^3_{E}(\theta, \lambda) = \frac{1}{n} \log \sum_{\{\delta_j, \sigma_i, 1 \leq j \leq m, 1 \leq i \leq n, S\}} \frac{(n_{\theta n}) (n_{\lambda n}) (3\theta - \lambda)}{(3\theta - \lambda)(3\theta - \lambda)} \frac{n!}{2^{\lambda n}} \frac{(3\theta - \lambda)^{\lambda n}}{(\lambda n)^{\lambda n}} \frac{3!}{3!} \frac{(\theta - \lambda)^n}{3^2} \times \left( \begin{array}{c} r \\ 0 \end{array} \right)^{\frac{(2\theta - \lambda - 3\theta n)}{2}} \left( \begin{array}{c} r \\ 1 \end{array} \right)^{\lambda n} \left( \begin{array}{c} r \\ 2 \end{array} \right)^{\frac{(3\theta - \lambda n)}{2}}
\]

(2.14)

Note that \( \frac{(3\theta - \lambda)n}{2} \) of even-valued \( \delta_i \)'s need to be 2 and \( \lambda n \) of \( \sigma_j \)'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 \( \delta_i \)'s and \( \sigma_j \)'s can be selected. The
total number of ways is \((\frac{(\lambda - \theta)n}{2})^{\theta n}\). For \(n \to \infty\), we simplify (2.14) using Stirling’s approximation, \(\log(n!) \approx n \log n - n\), and the binomial approximation, \(\log \left( \binom{n}{\mu n} \right) \approx n H_b(\rho, 1 - \rho)\), to obtain the result.

A special case of Corollary 1 is when \(\lambda \ll \theta\), i.e., when there exists only a small fraction of unsatisfied checks. In this case, we can approximate (2.13) as:

\[-2H_b(\theta, 1 - \theta) + \frac{3\theta}{2} \log \left( \frac{r}{2} \right) - H_b(\zeta, 1 - \zeta) + H_b \left( 1 - \zeta, \frac{3\theta}{2}, \frac{2\zeta - 3\theta}{2} \right), \tag{2.15}\]

which is exactly the same result as in Corollary 3.1 of [32] for the enumeration of elementary trapping sets. Therefore when \(\lambda \ll \theta\) and \(\theta\) is small, most (elementary) trapping sets satisfy the (elementary) absorbing set conditions. This observation is also shown in Fig. 2.6.

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

**Corollary 2.** The normalized logarithmic asymptotic distribution of \((a = \theta n, b = \lambda n)\) elementary absorbing sets in \(G_{n,m}^4\) is given by:

\[e^4_E(\theta, \lambda) = -3H_b(\theta, 1 - \theta) + \lambda \log 4r + \frac{4\theta - \lambda}{2} \log \left( \frac{r}{2} \right) + \theta H_b \left( \frac{\lambda}{\theta}, 1 - \frac{\lambda}{\theta} \right) - 4\theta H_b \left( \frac{\lambda}{4\theta}, 1 - \frac{\lambda}{4\theta} \right) - H_b(\zeta, 1 - \zeta) + H_b \left( 1 - \zeta, \lambda, \frac{4\theta - \lambda}{2}, \frac{2\zeta - 4\theta - \lambda}{2} \right). \tag{2.16}\]

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

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 3 considers this additional constraint and provides the asymptotic scaling of the average number of fully absorbing sets. Recall that for \(1 \leq j \leq n\), \(\delta_j\) denotes the number of edges that connect the \(j^{th}\) check node to the bit nodes in \(A_{a,b}^{(f)}\), and for
1 \leq i \leq a, \sigma_i \text{ denotes the number of edges that connect the } i^{th} \text{ bit node in the fully absorbing set } A_{a,b}^{(f)} \text{ to satisfied check nodes. Additionally, let } \mu_k, \text{ for } 1 \leq k \leq n - a, \text{ be the number of edges that connect the } k^{th} \text{ bit node from the subset of bit nodes not in the fully absorbing set (that is, for } 1 \leq k \leq n - a \text{) to the } (m - b) \text{ check nodes that themselves have even number of connections (including zero) to } A_{a,b}^{(f)}. \text{

Theorem 3. Let } 0 < \zeta, \theta, \lambda < 1 \text{ where } \zeta = \frac{m}{m} = \frac{r}{l}. \text{ Then, as } n \to \infty, p_{a,b,n}^{(f)}l,r \text{, the average number of } (a = \theta n, b = \lambda n) \text{ fully absorbing sets in } G_{n,m}^{l,r} \text{ scales as,}

\begin{equation}
p_{a,b,n}^{(f)}l,r \approx \sum_{\{\delta, \sigma, \mu: 1 \leq j \leq m, 1 \leq i \leq a, 1 \leq k \leq n - a, S, S_1\}} \left(\frac{n}{a}\right) \left(\frac{b}{m}\right) \prod_{i=1}^{a} \left(\frac{l}{\sigma_i}\right) \prod_{k=1}^{n-a} \left(\frac{r}{\mu_k}\right) \prod_{j=1}^{m} \left(\frac{\delta_j}{\delta}\right),
\end{equation}

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

\begin{equation}
S_1 = \left\{ \sum_{k=1}^{n-a} \mu_k = (m - b)r - q, \text{ for all } k \in \{1, 2, ..., n - a\}, \frac{l}{2} < \mu_k \leq l. \right\}
\end{equation}

Proof. The proof of the theorem extends the proof of Theorem 2. For a given } (a, b) \text{ fully absorbing set } A_{a,b}^{(f)}, \text{ express the transpose of the } m \times n \text{ parity check matrix } H \text{ as:

\[ H^T = \begin{bmatrix} M_1 & M_2 \\ M_{31} & M_{32} \end{bmatrix}, \]

where } M_1 \text{ and } M_2 \text{ are defined as before: the binary matrix } M_1 \text{ of size } a \times (m - b) \text{ (the binary matrix } M_2 \text{ of size } a \times b), \text{ corresponds to the subgraph spanned by the bit nodes in } A_{a,b}^{(f)} \text{ and the check nodes that are connected to } A_{a,b}^{(f)} \text{ even (odd) number of times. The matrix } M_{31} \text{ is an } (n - a) \times (m - b) \text{ binary matrix corresponding to the subgraph of the Tanner graph spanned by } (n - a) \text{ bit nodes not in } A_{a,b}^{(f)} \text{ and } (m - b) \text{ check nodes that are connected to the bit nodes in } A_{a,b}^{(f)} \text{ even number of times. Likewise, the matrix } M_{32} \text{ is an } (n - a) \times b \text{ matrix that corresponds to the subgraph spanned by } (n - a) \text{ bit matrices.}
\[ |\Lambda_{31}| = \frac{\left(\sum_{j=1}^{m-b} (r - \delta_j) \right)!}{\prod_{k=1}^{n-a} \mu_k! \prod_{j=1}^{m-b} (r - \delta_j)!} \cdot \exp \left[ -\frac{\sum_{k=1}^{n-a} \mu_k (\mu_k - 1) \sum_{j=1}^{m-b} (r - \delta_j)(r - \delta_j - 1)}{2 \left( \sum_{j=1}^{m-b} (r - \delta_j) \right)^2} \right] \] 

\[ (2.19) \]

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

\[ (2.20) \]

Nodes not in \( A_{a,b}^{(f)} \) and \( b \) check nodes that are themselves connected to the bit nodes in \( A_{a,b}^{(f)} \) odd number of times. Given the fully absorbing set constraints, \( \frac{1}{2} < \mu_k \leq l \) for \( 1 \leq k \leq n - a \). The values of \( \mu_k \)'s are also constrained by the \( \delta_j \)'s.

Following the discussion in the proof of Theorem 2, the quantities \(|\Lambda_1|\), \(|\Lambda_2|\), and \(|\Lambda_{n,m}^{l,r}|\) remain the same as in equations (2.7), (2.8) and (2.10). The original matrix \( M_3 \) is partitioned into two new matrices, \( M_{31} \) and \( M_{32} \). We define \( \Lambda_{31} \) and \( \Lambda_{32} \) matrix collections as:

\[ \Lambda_{31} = \{ \forall M \in \Lambda_{(n-a), (m-b)} : \text{For } 1 \leq k \leq n - a, \sum_{h=1}^{m-b} M(k, h) = \mu_k; \text{for } 1 \leq j \leq m - b, \sum_{g=1}^{n-a} M(g, j) = r - \delta_j \} \]

\[ \Lambda_{32} = \{ \forall M \in \Lambda_{n-a,b} : \text{For } 1 \leq k \leq n - a, \sum_{h=1}^{b} M(k, h) = l - \mu_k; \text{for } 1 \leq j \leq b, \sum_{g=1}^{n-a} M(g, j) = r - \delta_{m-b+j} \} \]

The asymptotic scaling of \(|\Lambda_{31}|\) and \(|\Lambda_{32}|\) are given in (2.19) and (2.20). Now, by using Lemma 1, we have

\[ z^{(f)l,r}_{a,b,n} = \sum_{\{\delta, \sigma, \mu_k : 1 \leq j \leq m, 1 \leq i \leq a, 1 \leq k \leq n-a, S, S_1\}} \binom{n}{a} \left(\begin{array}{c} m \\ b \end{array}\right) : |\Lambda_1| |\Lambda_2| |\Lambda_{31}| |\Lambda_{32}| \cdot (2.21) \]
Following the calculations in the proof of Theorem 1, as \( n \to \infty \), we have

\[
p^{(f)_{a,b,n}} \approx \sum_{\{\delta_j, \sigma_i, \mu_k:1 \leq j \leq m, 1 \leq i \leq a, 1 \leq k \leq n-a,S,S_1\}} \binom{n}{a} \binom{n}{b} \binom{m}{a} \binom{m}{b} \prod_{i=1}^{a} \frac{l}{\sigma_i} \prod_{k=1}^{n-a} \frac{l}{\mu_k} \prod_{j=1}^{r} \delta_j.
\]

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

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

\[
e^{(f)_{3,r}}_{E}(\theta, \lambda) = -2H_b(\theta, 1 - \theta) + \lambda \log 3r + \frac{3\theta - \lambda}{2} \log \left(\frac{r}{2}\right) + \theta H_b \left(\frac{\lambda}{\theta}, 1 - \frac{\lambda}{\theta}\right)
- 3\theta H_b \left(\frac{\lambda}{3\theta}, 1 - \frac{\lambda}{3\theta}\right) - H_b(\zeta, 1 - \zeta) + H_b \left(1 - \zeta, \lambda, \frac{3\theta - \lambda}{2}, \frac{2\zeta - 3\theta - \lambda}{2}\right) + \lambda(r - 1) \log 3
+ \left[(1 - \theta)H_b \left(\frac{\lambda(r - 1)}{1 - \theta}, 1 - \frac{\lambda(r - 1)}{1 - \theta}\right) - 3(1 - \theta)H_b \left(\frac{\lambda(r - 1)}{3(1 - \theta)}, 1 - \frac{\lambda(r - 1)}{3(1 - \theta)}\right)\right].
\]

(2.22)

**Proof.** Here, the constraints over \(\delta_j\)’s and \(\sigma_i\)’s are the same as in Corollary 1. 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 \(\mu_k\)’s are equal to 2 and \(n(1 - \theta) - (r - 1)\lambda n\) of \(\mu_k\)’s are equal to 3. By plugging in \(\delta_j\)’s, \(\sigma_i\)’s and \(\lambda_k\)’s in Theorem 3 and using the approximations as in Corollary 1, the result follows.

As \(\lambda \to 0\), the equation (2.22) reduces to (2.15), where in this regime, (elementary) trapping sets, absorbing sets, and fully absorbing sets are approximately the same objects. This observation is illustrated in Fig. 2.1. The part inside the brackets of
(2.22) comes from the additional constraints when an absorbing set is also a fully absorbing set. When \( \theta \) and \( \lambda \) 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 \( \lambda \) increases. It follows that the discrepancy between elementary absorbing sets and elementary fully absorbing sets increases as \( \lambda \) increases (shown in Fig. 2.1).

Using similar approximations, the following consequence of Theorem 3 readily follows.

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

\[
e^{(f)4,r}_E(\theta, \lambda) = -3H_b(\theta, 1 - \theta) + \lambda\log 4r + \frac{4\theta - \lambda}{2}\log \left(\frac{r}{2}\right) + \theta H_b \left(\frac{\lambda}{\theta}, 1 - \frac{\lambda}{\theta}\right) - 4\theta H_b \left(\frac{\lambda}{4\theta}, 1 - \frac{\lambda}{4\theta}\right) - H_b(\zeta, 1 - \zeta) + H_b \left(1 - \zeta, \lambda, \frac{4\theta - \lambda}{2}, \frac{2\zeta - 4\theta - \lambda}{2}\right) + \lambda(r - 1)\log 4
\]

\[
+ \left[(1 - \theta)H_b \left(\frac{\lambda(r - 1)}{1 - \theta}, 1 - \frac{\lambda(r - 1)}{1 - \theta}\right) - 4(1 - \theta)H_b \left(\frac{\lambda(r - 1)}{3(1 - \theta)}, 1 - \frac{\lambda(r - 1)}{3(1 - \theta)}\right)\right].
\]

(2.23)

Note that there exists a limit for the value of \( \lambda \) for absorbing sets to exist.

**Remark 1.** Consider the (fully) absorbing sets in the Tanner graphs in \( G^4_{n,m} \). 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 \( \lambda \leq 1 \) for \( b = \lambda n \). In addition, since \( 1 - R = \frac{1}{r} = \frac{m}{n} \), where \( R, R < 1 \), is the design rate, \( \lambda \) is also upper bounded by \( 1 - R \). The same argument follows for \( G^4_{n,m} \).

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 [37] trapping sets with
\[ \lambda \frac{n}{r} > 1, \text{ and therefore unstable, were considered in the analysis.} \]

## 2.3 Numerical Results

Fig. 2.1 compares the normalized logarithmic asymptotic distributions of elementary trapping sets and elementary (fully) absorbing sets for different \( G_{n,m}^{3,r} \), for \( \theta = 0.001 \). In this regime, elementary absorbing sets approximate absorbing sets well. As \( \lambda \) 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 (\( \Lambda_{3,n,m}^{3,6} \) vs. \( \Lambda_{3,n,m}^{3,15} \) vs. \( \Lambda_{n,m}^{3,30} \) collection), since having fewer checks of higher degrees makes it more difficult to meet additional combinatorial constraints.

![Figure 2.1: 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 \( \theta = 0.001 \).](image)

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 \( (3,6) \) and to \( (4,8) \)).
equal to $(3, 15)$ and to $(4, 20)$). There are significantly fewer absorbing sets of a particular size in $G_{n,m}^{4r}$ than in $G_{n,m}^{3r}$ 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: in the asymptotic limit the number of absorbing sets is subexponential for these choices of $\theta$ and $\lambda$.

![Figure 2.2](image)

Figure 2.2: Comparison of the normalized logarithmic asymptotic distributions of elementary absorbing sets in $G_{n,m}^{3r}$ and $G_{n,m}^{4r}$ for fixed $\theta = 0.001$, and for $\zeta$ 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.

Fig. 2.3 and Fig. 2.4 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 $\lambda$. 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 $\theta$ is different for different $\lambda$’s. Observe that in these two figures, for fixed $\theta$, $e^{1-r}(\theta, \lambda)$ increases as $\lambda$ increases for both trapping and absorbing sets. This agrees with the result from Fig. 2.1. We quickly remark that for $r < 9$ ($r \geq 9$), the curves are as those shown in Fig. 2.3 (Fig. 2.4). Thus for $r \geq 9$, there are exponentially many absorbing sets parametrized by given $\theta$ and $\lambda$, whereas for $r < 9$ there is a subexponential number of absorbing sets when $\theta$ and $\lambda$ are sufficiently small. Figures 2.3 and 2.4 also show that
as \( \theta \) increases or as \( \lambda \) decreases, trapping sets become better proxies for absorbing sets.

\[ h = [2e^{-4}, 4e^{-4}, 1e^{-3}, 2e^{-3}, 3e^{-3}, 5e^{-3}] \]

Figure 2.3: 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 \( \lambda \). The arrow indicates the increase in \( \lambda \) and the circles group up curves of the same \( \lambda \). The equations in [32] are used to plot the trapping set curves.

\[ h = [2e^{-4}, 4e^{-4}, 1e^{-3}, 2e^{-3}, 3e^{-3}, 5e^{-3}] \]

Figure 2.4: Comparison of the 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 \( \lambda \). The arrow indicates the increase in \( \lambda \) and the circles group up curves of the same \( \lambda \).

In Fig. 2.5 we plot the normalized logarithmic asymptotic elementary absorbing set distributions under the fixed ratio \( \eta = \frac{\lambda}{\theta} \) for various \( G_{n,m}^{3,f} \). Although the ratio of unsatisfied checks and absorbing set size is fixed, as \( \theta \) increases, the difference between
trapping sets and absorbing sets increases. It follows that trapping sets can better approximate absorbing sets for smaller $\theta$ when $\eta = 0.5$.

![Figure 2.5: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in the Tanner graphs in $G_{n,m}^{3,r}$ for the fixed ratio $\lambda/\theta = 0.5$. The equations in [32] are used to plot the trapping set curves.](image)

Following Fig. 2.5, Fig. 2.6 shows the normalized logarithmic asymptotic distributions of elementary trapping sets and absorbing sets in $G_{n,m}^{3,6}$ under different ratios $\eta = \lambda/\theta$. Note that the curves corresponding to small values of $\eta$ have the second zero crossing (see also Fig. 2.3). This second zero crossing can be used to represent the typical absorbing set distance (analogously to the typical trapping set distance, cf. [37]). It is also interesting to observe that the absorbing set curves taper off in Fig. 2.6. As $\eta$ increases towards 1, it again becomes more difficult to meet the combinatorial constraints of an absorbing set.

### 2.3.1 Implications for certain structured LDPC ensembles

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 [50], [51]. By comparing the results obtained
Figure 2.6: Comparison of the normalized logarithmic asymptotic distributions of elementary trapping sets (TS) and elementary absorbing sets (AS) in $C_{3,6}^{n,m}$ for different $\eta = \frac{\lambda}{\theta}$. Thicker lines correspond to increasing values of $\eta$, as the arrow indicates. The circles group up curves of the same $\eta$. The equations in [32] are used to plot the trapping set curves.

Figure 2.7: 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.

Here for the absorbing set analysis with the trapping set analysis of RAA codes presented in [38], 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 $\theta$ and $\lambda$. See Figure 2.7.

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

Effect of quantization parameters on the performance of BP decoder

3.1 Introduction

Low-density parity-check (LDPC) codes and accompanying message passing decoding algorithms are a popular choice for data encoding and decoding in modern communications and storage systems. To reduce implementation complexity, the messages in a practical message passing decoder are necessarily quantized. While it is well-known that the performance of practical message passing decoders in the high-reliability regime is governed by non-codeword decoding errors, a precise characterization of the relationship between such errors, the quantization choice and the overall performance is still missing.

Absorbing regions act as “decoding regions” around certain non-codeword fixed points known as absorbing sets. In this chapter, we take a closer look at the interplay between quantization and absorbing regions. We provide a study of a range of quantization choices, the impact of quantization on the candidate absorbing regions, and derive guidelines for practical decoders. We show that, due to the non-linear dynamics of message passing decoders, coarser quantization may in fact perform better than finer quantization. Results of this type can be particularly useful in designing high-
performance decoders for very high-reliability storage systems, such as emerging data storage hard disk and solid state drives.

3.2 BP Decoding

While maximum-likelihood decoding usually has a complexity exponential in the code length, BP decoding has a complexity linear in the code length. The BP decoding algorithm runs over the Tanner graph of an LDPC code and proceeds by applying Bayes’ rule locally and iteratively to update the messages over the edges of the graph [7]. Due to the presence of cycles in the Tanner graph, BP decoding is mathematically incorrect.

Let us consider a transmission over binary AWGN channel with a binary LDPC code. BP decoder computes the log-likelihood ratios of bits based on the Tanner graph of the code [52]. Suppose that there are \( n \) bit nodes and \( m \) check nodes. We let \( \mathcal{N}(i) \) denote the neighborhood set of a node \( i \). We consider BPSK modulation. The message exchange protocol is defined as follows [52]:

- The bit-to-check message \( m_{b\rightarrow c}(j, i) \), from the \( j \)th bit node to the \( i \)th check node, and the check-to-bit message \( m_{c\rightarrow b}(i, j) \), from the \( i \)th check node to the \( j \)th bit node, are initialized to zero for all bit nodes \( 1 \leq j \leq n \) and their neighboring check nodes \( i \in \mathcal{N}(j) \). Let \( \lambda_j \) denote the log-likelihood ratio of the \( j \)th bit at the channel output.

- \( k \)th iteration:

  Step 1: For the \( j \)th bit node, \( 1 \leq j \leq n \), calculate the message \( m_{b\rightarrow c}(j, i) \) to a neighboring check node as

  \[
  m_{b\rightarrow c}(j, i) = \lambda_j + \sum_{i' \in \mathcal{N}(j) \setminus \{i\}} m_{c\rightarrow b}(i', j). \tag{3.1}
  \]

  Step 2: For the \( i \)th check node, \( 1 \leq i \leq m \), calculate the message \( m_{c\rightarrow b}(i, j) \) to a
neighboring bit node as

\[ m_{c \rightarrow b}(i, j) = f^{-1} \left[ \sum_{j' \in N(i) \setminus \{j\}} f(|m_{b \rightarrow c}(j', i)|) \right], \quad (3.2) \]

where \( f(x) = -0.5 \log(\tanh(x/2)) \).

Step 3: For the \( j \)th bit node, \( 1 \leq j \leq n \), compute its posterior as

\[ \Lambda_j = \lambda_j + \sum_{i \in N(j)} m_{c \rightarrow b}(i, j). \quad (3.3) \]

A 0-1 bit decision (hard decision) is based on the sign of \( \Lambda_j \). If all such estimated bit values satisfy all of the parity check constraints, the currently decoded sequence is a codeword and the decoder terminates. Otherwise, the decoder proceeds with the next iteration until the maximum number of iterations is reached.

### 3.3 Quantization Parameters

In principle, \( m_{c \rightarrow b}(i, j) \) and \( m_{b \rightarrow c}(j, i) \) are some real numbers. In practical implementations, each message is quantized. It is convenient to use the notation \( Qa.b \) to denote the quantization of messages: parameter \( a \) stands for the integer part and parameter \( b \) stands for the fractional part of the messages. An additional bit is used to represent the sign of the message. Clearly, when the computed message exceeds the dynamic range established by \( a + b \), it is clipped accordingly. For example, \( Q0.0 \) corresponds to the bit flipping decoder which only has two levels for messages. It is obvious that for \( Qa.b \), we are using \( a + b + 1 \) bits to represent each message in the decoder. As we increase the values of \( a \) and \( b \), the complexity of the computations increases. Therefore, it is desirable to use fewer bits in quantization to reduce the complexity of the decoder.
3.4 Experimental results

We performed several experiments over a range of code parameters of array-based codes, and made the following observations:

- Whenever $a$ was 3 or less, $Q_{a,b}$ performed better than $Q_{a,b'}$ for $b < b'$.
- Whenever $a$ was 4 or more, $Q_{a,b}$ performed better than $Q_{a,b'}$ for $b > b'$.
- For $a, a' \geq 4$ and distinct, for $a + b = a' + b'$, $Q_{a,b}$ and $Q_{a',b'}$ have different absorbing set profiles even when the overall FERs are comparable.

These observations are illustrated in part in Figures 3.1 and 3.2 and for the (2212, 1896) array-based code (code length $n = 2212$ with 1896 input information bits).

In figure 3.1, when $a = 2$, we can see that for the coarser quantization parameters, i.e., smaller values of $b$, the decoder performs better and the BER reduces. In figure 3.2, where $a + b$ is 6 for all curves, we can see that there is a dramatic improvement in performance of the decoder as we change $a$ from 3 to 4. On the other hand, there not much change in performance of the decoder when we change parameter $a$ from 3 to 4. In this case, although the performance is almost identical, the error profile changes.

**Discussion:** We contribute the performance variation to the changes in the attractor regions around critical absorbing sets.
Figure 3.2: Performance comparison of the decoder for $a = 3, 4, 5$ and $a + b = 6$.

Figure 3.3: A topological overlap of $(6, 4)$ and $(8, 2)$ absorbing sets. Circles denote variable nodes and squares denote check nodes. Assume that circles in an absorbing set represent bit nodes with value ‘1’ and all other bit nodes have values ‘0’. The four unsatisfied checks under the $(6, 4)$ absorbing set (marked with diagonal lines) become satisfied with the addition of two carefully placed bit nodes. With these eight bit nodes, two checks are unsatisfied.
For $Q_{4.2}$ and $Q_{5.1}$, while the overall performance is comparable, there exist noise configurations that cause certain absorbing set errors under $Q_{4.2}$ but not under $Q_{5.1}$, and vice versa. In particular, $(6, 4)$ absorbing set errors dominate the performance under $Q_{4.2}$. However, certain such errors either get corrected altogether or become $(8, 2)$ absorbing set errors under $Q_{5.1}$. The key to this shift is the critical topological overlap between a $(6, 4)$ and a $(8, 2)$ absorbing set, as shown in Figure 3.3.

Such insights can be particularly useful when designing practical decoders. For example, one may want to switch to a different quantization scheme of equal or lower complexity in later iterations for the purposes of escaping the absorbing set errors. The absorbing region viewpoint aids in a systematic development of such an approach.
Chapter 4

Conclusion

In this thesis, the normalized logarithmic asymptotic distributions of absorbing sets and fully absorbing sets for regular LDPC code ensembles was computed. 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. Other results show that for small \(\theta\) and \(\lambda\), 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 \(\theta\) and \(\lambda\) are small and when the ratio \(\frac{\lambda}{\theta}\) is smaller 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. Also, an unexpected and interesting observation on effect of quantization parameters of performance of message-passing decoder is demonstrated. Empirical results was provided to show that coarser
quantization sometimes performs better than finer quantization.
Bibliography


[10] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, “A class of low-density parity-
check codes constructed based on Reed-Solomon codes with two information 

of quasi-cyclic LDPC codes for the AWGN and erasure channels,” *IEEE Trans. 


[14] J. Pearl, “Reverend Bayes on inference engines: a distributed hierarchical ap-
proach,” in *Proceedings of American Association for Artificial Intelligence Na-


codes with a large number of short cycles” in *Proceedings of IEEE Vehicular 

[18] V. Zyablov and M. Pinsker, “Estimation of the error correction complexity of Gal-


