BOUNDS ON THE MAP THRESHOLD OF ITERATIVE DECODING SYSTEMS WITH ERASURE NOISE

A Thesis

by

CHIA-WEN WANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2008

Major Subject: Electrical Engineering

BOUNDS ON THE MAP THRESHOLD OF ITERATIVE DECODING SYSTEMS WITH ERASURE NOISE

A Thesis

by

CHIA-WEN WANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Henry D. Pfister
Committee Members,	Krishna R. Narayanan
	Jim Ji
	Harold P. Boas
Head of Department,	Costas N. Georghiades

August 2008

Major Subject: Electrical Engineering

ABSTRACT

Bounds on the MAP Threshold of Iterative Decoding Systems with Erasure Noise. (August 2008)

Chia-Wen Wang, B.S., National Cheng Kung University

Chair of Advisory Committee: Henry D. Pfister

Iterative decoding and codes on graphs were first devised by Gallager in 1960, and then rediscovered by Berrou, Glavieux and Thitimajshima in 1993. This technique plays an important role in modern communications, especially in coding theory and practice. In particular, low-density parity-check (LDPC) codes, introduced by Gallager in the 1960s, are the class of codes at the heart of iterative coding. Since these codes are quite general and exhibit good performance under message-passing decoding, they play an important role in communications research today.

A thorough analysis of iterative decoding systems and the relationship between maximum *a posteriori* (MAP) and belief propagation (BP) decoding was initiated by Méasson, Montanari, and Urbanke. This analysis is based on density evolution (DE), and extrinsic information transfer (EXIT) functions, introduced by ten Brink.

Following their work, this thesis considers the MAP decoding thresholds of three iterative decoding systems. First, irregular repeat-accumulate (IRA) and accumulate-repeat-accumulate (ARA) code ensembles are analyzed on the binary erasure channel (BEC). Next, the joint iterative decoding of LDPC codes is studied on the dicode erasure channel (DEC). The DEC is a two-state intersymbol-interference (ISI) channel with erasure noise, and it is the simplest example of an ISI channel with erasure noise. Then, we introduce a slight generalization of the EXIT area theorem and apply the MAP threshold bound for the joint decoder. Both the MAP and BP erasure

thresholds are computed and compared with each other. The result quantifies the loss due to iterative decoding

Some open questions include the tightness of these bounds and the extensions to non-erasure channels.

To my parents

ACKNOWLEDGMENTS

My deepest gratitude goes first to my parents and brother, for their love and for full support for my study here.

I would like to express my gratitude to my advisor, Dr. Henry D. Pfister, for his guidance, supervision, and encouragement. I especially appreciate the opportunity to work with him and receive many valuable insights from him in research and life.

I would like to thank my committee members, Dr. Narayanan, Dr. Boas and Dr. Ji, for great support to me during the days of my thesis preparation.

I am also grateful to my girlfriend, Yuli, for her emotional support and editorial assistance to my thesis writing.

I would also like to thank office mates, Fan, Byung Hak, Wei-Yu, Phong, Arvind, Yung-Yih, and other friends and Professors, for all kind of support in WCL.

Lastly, my thanks go to other family and friends who have provided help while I studied in the United States.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

In communications, the main goal is the transmission of a message across a noisy channel (phone line, optical link, wireless, ...) so that the receiver can determine this message with high probability despite the imperfections of the channel. How can we efficiently and reliably transmit information? When Shannon published his seminal paper "A Mathematical Theory of Communication" [1], he gave the basic answer: coding can do it. That is, add redundancy to the message that can be exploited to combat the distortion introduced by the channel. After Shannon, the search for practical coding systems that approach this fundamental limit established by Shannon has been at the heart research in communications.

In the area of digital communications, an error-correcting code (ECC) or forward error correction (FEC) code is a set of signals chosen so that each data signal conforms to specific rules of construction. Therefore, departures from this construction in the received signal can generally be automatically detected and corrected (i.e., to reconstruct the original, error-free data). In addition, ECCs are used to represent sources efficiently, maintain data integrity across noisy channels, avoid large peak-to-average power ratios, and minimize interference in multi-user systems.

From the coding perspective, iterative decoding and codes on graphs was first devised by Gallager in 1960 [2], and then rediscovered by Berrou, Glavieux and Thitimajshima in 1993 [3]. This technique plays an important role in modern communications, especially in coding theory and practice. In particular, low-density parity-check (LDPC) codes, introduced by Gallager in the 1960s, are the class of codes at the heart

---

The journal model is *IEEE Transactions on Automatic Control.*

of iterative-coding idea. LDPC codes are linear codes which can be defined in terms of a sparse parity-check matrix. The main reason for focusing on these codes is that they are quite general and their performance under message-passing decoding is quite good.

A thorough analysis of iterative decoding systems and the relationship between maximum *a posteriori* (MAP) and belief propagation (BP) decoding was initiated by Méasson, Montanari, and Urbanke in [4], [5]. This analysis is based on density evolution (DE) and extrinsic information transfer (EXIT) functions [6]. Their work focuses mainly on LDPC and turbo codes, but they note that these ideas can be extended to other iterative decoding systems. In this thesis, we extend some of their results to irregular repeat-accumulate (IRA), and accumulate-repeat-accumulate (ARA). Then, we introduce a slight generalization of the EXIT area theorem and apply it to the joint iterative decoding of LDPC codes over channels with memory.

DE is a method of evaluating iterative decoding systems for asymptotically large block lengths and was introduced in [7]. EXIT functions were introduced by ten Brink as an approximate technique to visualize the convergence of iterative systems [6]. In fact, for the erasure channel, EXIT functions satisfy a rigorous conservation law known as the area theorem [8]. The area theorem can be used to rigorously connect the performance of a code under MAP decoding to its performance under BP decoding. Méasson, Montanari and Urbanke gave a graphical construction of the MAP threshold using an approach reminiscent of the Maxwell construction in thermodynamics to provide a bridge between MAP and BP decoding [4], [5].

Jin, Khandekar, and McEliece proposed and analyzed IRA codes in [9]. ARA codes were introduced by Abbasfar, Divsalar, and Kung in [10]. Later, it was shown that the DE analysis of IRA and ARA codes can be reduced to the DE analysis of LDPC codes via a technique known as graph reduction [11].

The idea of decoding a code transmitted over a channel with memory via iteration was first introduced by Douillard, *et al.* in the context of turbo codes and is known as *turbo equalization* [12]. Turbo equalization can also be extended to the joint decoding of LDPC codes by constructing one large graph which represents the constraints of both the channel and the code [13]. For finite-state (FS) channels, analysis of joint decoding requires the analysis of the BCJR algorithm which is used to decode the channel. For some channels, DE can be done analytically for the joint iterative decoding of irregular LDPC codes and the channel [14]. One such channel is the dicode erasure channel (DEC), which is simply a binary-input channel with a linear response of $1 - D$ and erasure noise.

In this thesis, we consider the erasure threshold of MAP decoding for a code ensemble (i.e., the erasure threshold where the average entropy $h^{MAP}(\epsilon)$ of a code converges to zero when the erasure probability is less than the threshold). We apply the ideas of [4], [5], [15] (extend the Maxwell construction approach and use the upper bounding technique) to IRA ensembles, ARA ensmbles, and the joint iterative decoding of irregular LDPC codes and the DEC (channel with memory). Both the MAP and BP erasure thresholds are computed and compared with each other.

In Chapter II, a brief background is given for iterative decoding, LDPC codes, density evolution, EXIT functions, area theorem, and the MAP threshold bounding technique.

In Chapter III, we give the background on IRA and ARA codes and density evolution and fixed point analysis of iterative decoding for IRA and ARA codes. The MAP threshold bounding technique is applied to IRA and ARA codes. Finally, the tightness of the bound is discussed.

In Chapter IV, we give a system description and briefly introduce density evolution for joint iterative decoding. Then, the MAP threshold bounding technique is applied to joint decoding.

In Chapter V, concluding remarks and open questions are discussed.

CHAPTER II

LOW-DENSITY PARITY-CHECK CODES

Low-density parity-check (LDPC) codes are linear codes which have a sparse graph representation; in general, they tend to exhibit good performance under message-passing decoding. This chapter gives a summary of well-known results that will be used in later chapters [16].

A. Regular LDPC Codes

A (d_v, d_c)-regular LDPC code is a binary linear code such that every bit node has degree d_v and every check node has degree d_c.

Let n be the length of the binary code and a bipartite graph with bit nodes and check nodes is given, respectively, by

$$n \quad and \quad m \triangleq \frac{nd_v}{d_c}.$$

Every codeword x is a solution to the parity-check equation of binary code with length n is given by $Hx^T = 0^T$. Each bit node corresponds to one bit of the codeword (i.e., one column of H), and each check node corresponds to one parity-check equation (i.e., one row of H). As a nonzero entries of H, edges in Fig. 1, connecting bit nodes to check nodes, are in one-to-one correspondence.

There are $nd_v = md_c$ edges in the bipartite graph where d_v edges correspond to each bit node on the top and d_c edges correspond to each check nodes on the bottom. Fig. 1 gives an example of a (3,6)-regular code of length 6.

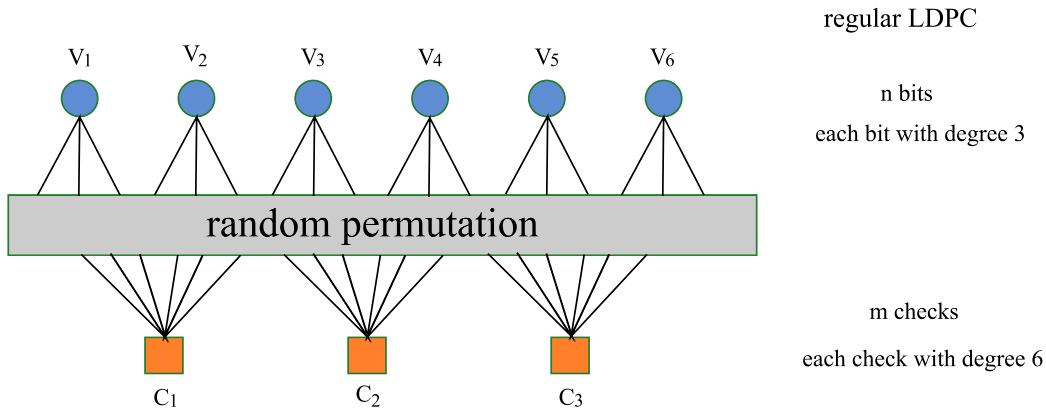

Fig. 1. A (3,6)-regular code of length 6. There are 6 bit nodes and 3 check nodes. There are $nd_v = 18 = md_c$ "sockets" of the bit and check nodes.

B. Degree Distribution

An irregular LDPC ensemble is described by its degree distribution (d.d.), which encodes the fraction of nodes (or edges) with a particular degree.

From an edge perspective, the degree distribution of the bit and check nodes is given, respectively, by

$$\lambda(x) = \sum_{i=1}^{\infty} \lambda_i x^{i-1} \quad and \quad \rho(x) = \sum_{i=1}^{\infty} \rho_i x^{i-1},$$

where λ_i is the fraction of edges that are adjacent to an bit node of degree i, and ρ_i is the fraction of such edges that are adjacent to an check node of degree i.

From a node perspective, the degree distribution of bit and check nodes is given, respectively, by

$$L(x) = \sum_{i=1}^{\infty} L_i x^i \quad and \quad R(x) = \sum_{i=1}^{\infty} R_i x^i,$$

where L_i represent fraction of bit nodes of degree i, and R_i represent fraction of check nodes of degree i. .

The design rate of the code in terms of its degree distribution is given by

$$r_{LDPC} = 1 - \frac{L'(1)}{R'(1)} = 1 - \frac{\int_0^1 \rho(x)\,dx}{\int_0^1 \lambda(x)\,dx}.$$

The design rate is the rate of the code assuming that all constraints are linearly independent (i.e., the code's true rate r may be higher due to linear dependence in the parity-check matrix).

C. Ensembles

Given a degree distribution pair (λ, ρ), the ensemble of LDPC codes of length n is defined as follows. Each graph in LDPC (λ, ρ) has $nL'(1)$ variable nodes and $nrR'(1)$ check nodes. A node of degree i has i sockets from which the i edges emanate, so that in total there are $nL'(1) = nrR'(1)$ sockets on each side. Label the sockets on each side with the set $[nL'(1)] \triangleq \{1, \ldots, nL'(1)\}$ in some arbitrary but fixed way. Let π be a permutation on $[nL'(1)]$. With π, in a bipartite graph, the i-th socket on the bit side to the $\pi(i)$-th socket on the check side. Letting π run over the set of permutations on $[nL'(1)]$ generates a set of bipartite graphs. Finally, a probability distribution over the set of graphs is defined by placing a uniform probability distribution on the set of permutations. This is the ensemble of bipartite graphs LDPC codes (n, λ, ρ).

Codes in this thesis are chosen randomly from an ensemble by choosing a random permutation to connect the bit and check nodes [17, p. 579], [18].

D. Message-Passing Decoder

First, we briefly introduce a message-passing algorithm to accomplish the decoding process. Basically, the algorithm proceeds by sending message from one node to the other. For the classes of codes that we will consider in the following chapter, the messages-passing with bit node and check node is showed in Fig. 2.

Fig. 2. Message-passing rules with bit node as a circle and check node as a square.

From the left in Fig. 2, message sent along the edge from the check node to the bit node is

$$\nu_i^{l+1} = f(\mu_1^l,\, \mu_2^l,\, \ldots,\, \mu_{i-1}^l,\, \mu_{i+1}^l,\, \ldots,\, \mu_{d_c}^l),$$

where ν_i^{l+1} is the message passed from check node to bit node for degree type i and iteration $l+1$. μ_i^l is the message passed from bit node to check node for degree type i and iteration l.

From the right in Fig. 2, message sent along the edge from the bit node to the check node is

$$\mu_i^{l+1} = g(\nu_1^l,\, \nu_2^l,\, \ldots,\, \nu_{i-1}^l,\, \nu_{i+1}^l,\, \ldots,\, \nu_{d_c}^l).$$

For the case of transmission over a binary channel, the messages can be compressed to a single real quantity. Particularly, if we choose this quantity to be the log-likelihood ratio (LLR) (log of the ratio of the two likelihoods [16, p. 56]) then the processing rules can be described as: function g denotes messages add at the bit node, and function f denotes at check nodes the processing rule is stated in [16, p. 57, (2.16)].

With the message-passing algorithm that we discussed above, we further specialize this algorithm for binary erasure channel (BEC).

1. Decoding on the Erasure Channel

The output messages through BEC corresponds to the three possibilities, namely that the received values are 0, ? (erasure), or 1, respectively. With the three possible outputs, the general message-passing rules for the BEC is discussed as the following: at a bit node the outgoing message is an erasure if all incoming message are erasures. Therefore, the outgoing message is equal to this common value. Otherwise, all non-erasure messages must agree and either be 0 or 1 because of no error is introduced in the channel. Also, at a check node the outgoing message is an erasure if any of the incoming message is an erasure. Otherwise, if all of the incoming message are either 0 or 1 then the outgoing message is the XOR of all the incoming message. That said, for the BEC message-passing allows us to find an yet unknown value from already known ones by iteratively checking the corresponding parity-constraints.

Fig. 3 shows the process of the message-passing decoder to the simple irregular LDPC code assuming that the received word is (0, ?, 1, ?, 1). In iteration 0 (this is shown on the right of Fig. 3 in the first from the top figure) the bit-to-check messages correspond to the received values. Next, consider the check-to-variable message sent in iteration 1 from check node 1 to bit node 2 (left side of Fig. 3 in the second from the top figure). This message is 1 (the mod-2 sum of incoming messages) according to

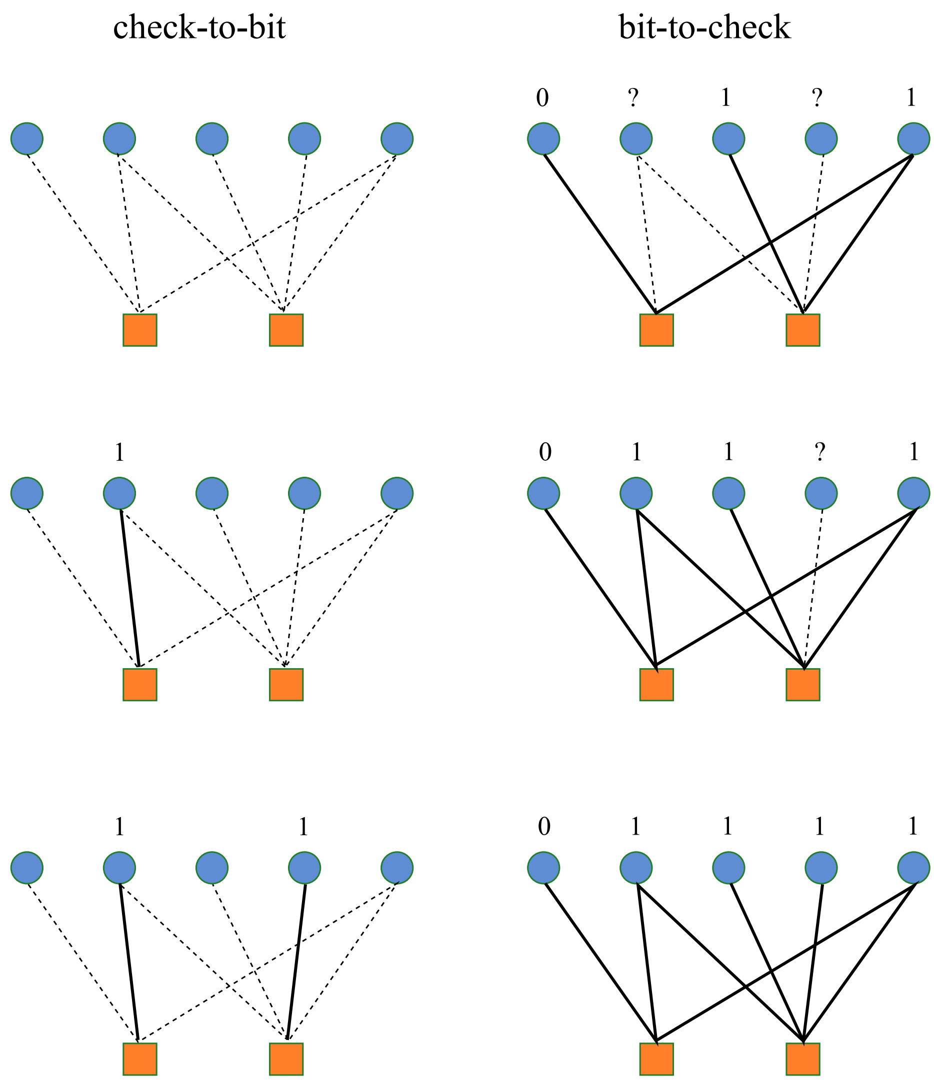

Fig. 3. Message-passing decoding of the the simple irregular LDPC code with the received word (0, ?, 1, ?, 1). The three rows correspond to iterations 0 to 2. After the first iteration we recover $x_2 = 1$, after the second we know that $x_4 = 1$. This means that for this case the recovered codeword is (0, 1, 1, 1, 1).

the message-passing rule. This message reflects the fact that through the parity-check constraint $x_1+x_2+x_5=0$ we can find x_2 given x_1 and x_5. Then, x_2 passes the "new" value with other already known ones to check nodes to move on the greedily iterative decoding process. Finally, after two iterations the transmitted word is found to be (0, 1, 1, 1, 1).

E. Concentration

The concentration theorem below states that the individual elements of an ensemble behave with high probability close to the ensemble average.

Theorem 1 (Concentration Around Ensemble Average [16]). Let the code G, chosen uniformly at random from LDPC $(n,\, \lambda,\, \rho)$, be used for transmission over the BEC (ϵ) (output are erased with probability ϵ). Assume that the decoder performs l rounds of message-passing decoding and let $P_b^{BP}(G,\, \epsilon,\, l)$ denote the resulting bit erasure probability. Then, for l fixed and for any given $\delta > 0$, there exists an $\alpha > 0$, $\alpha = \alpha\,(\lambda,\, \rho,\, \epsilon,\, \delta,\, l)$, such that

$$P\left\{|P_b^{BP}(G,\, \epsilon,\, l) - \mathbb{E}_{G' \in LDPC(n,\, \lambda,\, \rho)}\left[P_b^{BP}(G,\, \epsilon,\, l)\right]| > \delta\right\} \leq e^{-\alpha n}.$$

That is, the theorem asserts that all except an exponentially (in the blocklength) small fraction of codes behave within an arbitrarily small δ from the ensemble average. For sufficiently large blocklength n, the ensemble average is a good indicator for the individual behavior. Therefore, it is sufficient to consider the average behavior.

F. Density Evolution

The performance of irregular LDPC codes can be significantly better than regular LDPC codes. Certain structural modifications, such as those provided by IRA and ARA constructions can also improve performance. DE can be used to analyze and design (e.g., optimize the degree distribution) LDPC, IRA, and ARA codes. DE works by recursively tracking the distribution of messages passed around the Gallager-Tanner-Wiberg (GTW) graph during iterative decoding. It also gives a precise characterization of the asymptotic performance in terms of a noise threshold, where decoding almost surely converges if the noise is less than the threshold.

Theorem 2 (Density Evolution). Consider a degree distribution pair (λ, ρ) through a BEC with erasure probability ϵ. The DE recursion can be written in closed form as

$$x_{l+1} = \epsilon\lambda\left(1 - \rho\left(1 - x_l\right)\right),$$

where x_l is the average fraction of erasure messages sent from the bit nodes to the check nodes during iteration l.

Proof. While we consider x_l first, the initial bit-to-check message is equal to the received message which is an erasure message with probability ϵ. Then, it follows that $x_l = \epsilon$. Next, consider x_{l+1}. We start with the check-to-bit messages in the $(l + 1)$-th iteration. According to the message-passing algorithm, a check-to-bit message emitted by a check node of degree i along a particular edge is the erasure message if any of the $(i - 1)$ incoming messages is an erasure. Since it is assume that each message is an erasure with probability x_l and all messages are independent, the probability that the outgoing message is an erasure is equal to $1 - (1 - x_l)^{i-1}$. Since the edge has probability ρ_i to be connected to a check node of degree i, it follows that the expected erasure probability of a check-to-bit message in the $(l+1)$-th iteration is equal to $\sum_i \rho_i \left(1-(1-x_l)^{i-1}\right) = 1-\rho(1-x_l)$. Similarly, we consider the erasure probability of the bit-to-check messages in the $(l+1)$-th iteration. Consider an edge e which is connected to a bit node of degree i. The outgoing bit-to-check message along this edge in the $(l+1)$-th iteration is an erasure if the received value of the associated variable node is an erasure and all $(i-1)$ incoming messages are erasure. This comes with probability $\epsilon(1-\rho(1-x_l))^{i-1}$. Averaging over the edge degree distribution λ, we get the DE in closed form as $x_{l+1} = \epsilon\lambda(1-\rho(1-x_l))$. □

Remark 1. The DE technique was introduced in [18]. The main assumption of DE is that the message passed on the edges of the Tanner graph are statistically independent. This assumption is justified by the fact that, for randomly chosen codes, the fraction of bits involved in finite-length cycles vanishes as the block length tends to infinity.

Remark 2. Given a degree distribution pair $(\lambda,\, \rho)$ and a real number ϵ , $\epsilon \in \left[0,\, \epsilon^{BP}\right]$ (ϵ^{BP} is the erasure threshold of BP decoding for a code ensemble), if x_{l+1} is less than $\epsilon\lambda(1-\rho(1-x_l))$ for $x \in (0,\, 1]$, it will result in the convergence behavior (see Fig. 4). That is, the condition for convergence can hence be written as

$$x_{l+1} < \epsilon\lambda(1-\rho(1-x_l))\,,\, x \in (0,\, 1].$$

G. EXIT Functions and the Area Theorem

EXIT functions first appeared as handy tools to visualize the iterative decoding process; from EXIT curves, one can easily see the "bottlenecks" in the iterative decoding process [6]. Once these critical regions have been identified, the component codes can

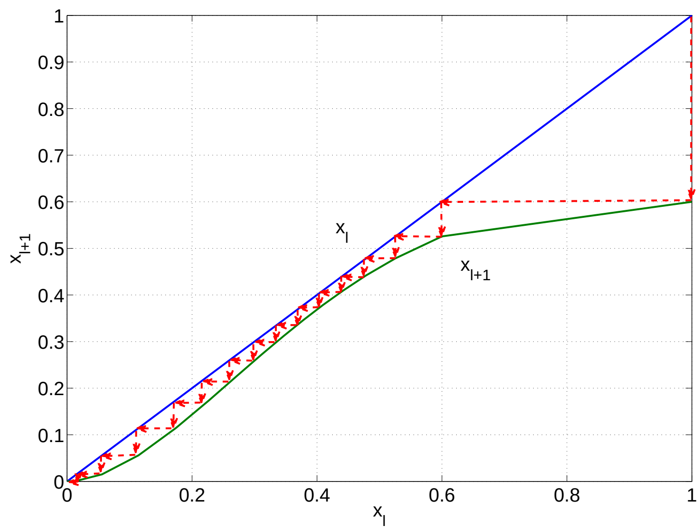

Fig. 4. The evolution of the decoding process for the dd pair $(\lambda(x), \rho(x)) = (x^2, x^3)$, and $\epsilon = 0.6$. The initial fraction of erasure messages emitted by the bit nodes is $x_l = 1$. After an iteration (at the next output of the bit nodes) this fraction has evolved to $x_{l+1} = 0.6$. After second full iteration, i.e., at the output of the bit nodes we see an erasure fraction of $x = 0.5257$. This process continues in the same fashion for each subsequent iteration, corresponding graphically to a staircase function which is bounded below by x_l and bounded above by x_{l+1}.

be changed appropriately to "match" the curves and improve the performance of the system.

Definition 1. Let $\mathcal{C}$ be a length-n binary code defined by the probability distribution $p_{X_1^n}(x_1^n)$. Let X_1^n be chosen according to $p_{X_1^n}(x_1^n)$ and Y_1^n be the result of transmitting X_1^n over a BEC(ϵ). Then, the *MAP EXIT function* is defined to be

$$h^{MAP}(\epsilon) \triangleq \frac{1}{n} \sum_{i=1}^{n} H\left(X_i | Y_1^n(\epsilon) \setminus Y_i(\epsilon)\right).$$

Remark 3. From this, we see that $h^{MAP}(\epsilon)$ is the average (over all bits) entropy of the optimal *a posteriori* probability (APP) estimate of X_i from the observations Y_1^n except Y_i. The notation $Y_1^n(\epsilon)$ and $Y_i(\epsilon)$ is used to emphasize the dependence of these r.v. on ϵ. Let ϵ^{MAP} be the erasure threshold of MAP decoding for a code ensemble. For asymptotically large n, the average conditional entropy $h^{MAP}(\epsilon)$ converges to zero for $\epsilon < \epsilon^{MAP}$ and is strictly positive for $\epsilon > \epsilon^{MAP}$.

Theorem 3 (Area Theorem). Let $\mathcal{C}$ be a length-n binary code defined by the probability distribution $p_{X_1^n}(x_1^n)$. Let X_1^n be chosen according to $p_{X_1^n}(x_1^n)$ and Y_1^n be the result of transmitting X_1^n over a BEC(ϵ). To emphasize that Y_1^n depends on the channel parameter δ write $Y_1^n(\delta)$. Then

$$\frac{1}{n}H\left(X_1^n|Y_1^n(\delta)\right) = \int_0^\delta h^{MAP}(\epsilon)\, d\epsilon.$$

Proof. A nice history of this theorem and its various proofs can be found in [5, p. 44]. □

In addition, there is another, perhaps more surprising, application of EXIT functions; they can be used to connect the performance of a code under BP decoding to that under MAP decoding.

Definition 2. The *BP EXIT function* of a length-n code is given by

$$h^{BP}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} h_i^{BP}(\epsilon),$$

where $h_i^{BP}(\epsilon)$ is the entropy of the iterative decoding estimate of X_i from Y_1^n except Y_i. The iterative decoding estimate of a bit is given by the bit's extrinsic message in the BP decoder after l iterations of decoding.

Remark 4. For ensembles of codes, these expressions will also refer to the asymptotic EXIT functions as $n \to \infty$. In the case of BP EXIT functions, we assume also that the number of decoding iterations $l \to \infty$ as well (with the n-limit taken first). These limits are well-defined and deterministic for BP EXIT functions because of the concentration theorem [18]. For the MAP EXIT function, a similar approach can be used to show that $h^{MAP}(\epsilon)$ concentrates around its ensemble average [19] which we will assume to be well-defined.

The following parametric expression for the asymptotic BP EXIT function is given in [4], [5] for the standard ensemble of LDPC codes.

Theorem 4. [16] For an irregular LDPC code, the asymptotic BP EXIT curve is given in parametric form by

$$h^{BP}(\epsilon(x)) = \begin{cases} 0, & x \in [0, x^{BP}) \leftrightarrow \epsilon \in [0, \epsilon^{BP}) \\ L(1-\rho(1-x)) & x \in (x^{BP}, 1] \leftrightarrow \epsilon \in (\epsilon^{BP}, 1] \end{cases},$$

where $\epsilon(x) \triangleq \frac{x}{\lambda(1-\rho(1-x))}$, and x^{BP} denotes the location of the unique minimum of $\epsilon(x)$ in the range (0,1] and $\epsilon^{BP} \triangleq \epsilon(x^{BP})$ is the BP decoding threshold.

The BP EXIT function $h^{BP}(\epsilon)$ of a (3,6)-regular LDPC code on the erasure channel is shown in Fig 5. Its BP threshold ϵ^{BP} is given by the point where $h^{BP}(\epsilon)$ drops down to zero.

H. Bounding the MAP Decoding Threshold

The following approach to bounding the MAP decoding threshold is based on the approach used in [4], [5]. The key point is that the optimality of the MAP decoder implies

$$h^{MAP}(\epsilon) \leq h^{BP}(\epsilon).$$

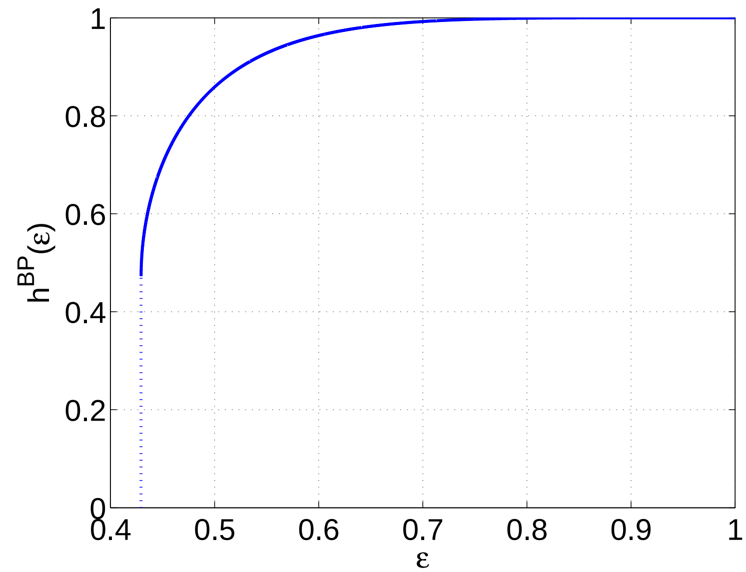

Fig. 5. The BP EXIT function $h^{BP}(\epsilon)$ of a (3,6)-regular LDPC code on the erasure channel. The BP threshold ϵ^{BP} is given by the point where $h^{BP}(\epsilon)$ drops down to zero.

Since the integral of $h^{MAP}(\epsilon)$ is equal to the code's true rate r (based on the area theorem), it follows that

$$r_{LDPC} \leq r = \int_{\epsilon^{MAP}}^{1} h^{MAP}(\epsilon)\, d\epsilon \leq \int_{\epsilon^{MAP}}^{1} h^{BP}(\epsilon)\, d\epsilon$$

because $h^{MAP}(\epsilon) = 0$ for $0 \leq \epsilon \leq \epsilon_{MAP}$ and $r_{LDPC} \leq r$ (i.e., linear dependencies in the parity check-matrix can only increase the rate). This bound is useful because $h^{BP}(\epsilon)$ can be computed easily. In some cases, it can also be shown that the bound is tight and that $h^{MAP}(\epsilon) = h^{BP}(\epsilon)$ for $\epsilon > \epsilon^{MAP}$ [4], [5].

Fig. 6 shows the BP EXIT function $h^{BP}(\epsilon)$ and the integral bound on ϵ^{MAP}. In this construction, the left edge of the shading is chosen so that the shaded area under the BP EXIT function curve equals the code rate. This left edge provides the upper bound on the MAP threshold ϵ^{MAP}. In addition, Fig. 7 shows the integral process in Fig. 6 where $I(\epsilon) = \int_{\epsilon}^{1} h^{BP}(x)\, dx$.

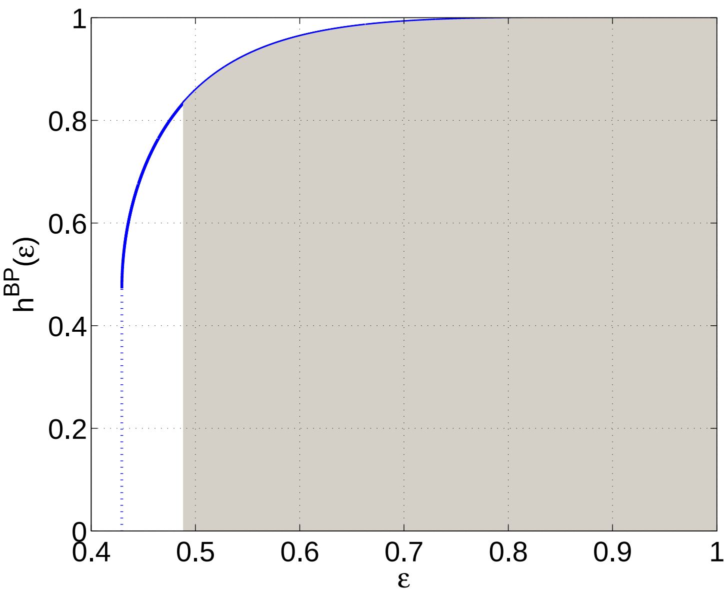

Fig. 6. The BP EXIT function $h^{BP}(\epsilon)$ of a (3,6)-regular LDPC code on the erasure channel. The left boundary of the shaded area is the upper bound on ϵ^{MAP}, and the area of the shaded portion under the curve equals the code rate of $\frac{1}{2}$. This gives $\epsilon^{BP} = 0.4294$ and $\epsilon^{MAP} \leq 0.4881$. (Note: $\epsilon^{Shannon} = 0.5$)

I. Peeling Decoder and Residual Graph

The peeling decoder has identical performance to the message-passing decoder. Both decoders get stuck in the largest "stopping set" which is included in the set of erased bits. A stopping set is a subset of bit nodes together whose set of neighboring check nodes includes no checks of degree one. Also, the peeling decoder gets stuck in exactly the same structure. Therefore, with an infinite number of iteration, the performance of the two decoders is identical.

The computation rules of the peeling decoder differ from those of the message-

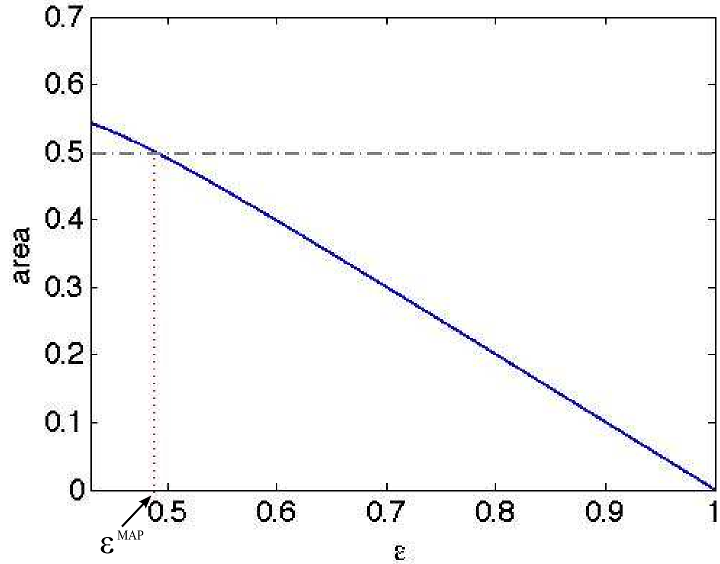

Fig. 7. The integral curve shows the integral process in Fig. 6 where $I(\epsilon) = \int_{\epsilon}^{1} h^{BP}(x)\,dx$. The curve gradually increases as ϵ is decreased from 1. When the integral curve intercepts the horizontal 0.5 dot line, the result is an upper bound on ϵ^{MAP}.

passing one in two aspects: (i) at the bit nodes we do not obey the message-passing principle but we replace the received value with the current estimate of the bit based on all incoming messages and the received message; (ii) rather than updating all messages in parallel we pick in each step one check node and update its outgoing messages as well as the messages of its neighboring bit nodes.

In addition, without changing the behavior of the algorithm, we further apply the following simplifications : once a non-erasure message has been sent out along a check node (i.e., this check node has served its purpose), it no longer plays a role in the future decoding. A check node sends out a non-erased message unless all but possibly one of its neighbors are known. Therefore, after processing this check node all its neighbors are known, we can safely delete from the graph any such check node and all its attached edges. Similarly, each known bit node can send to its neighboring check node its value and these values are accumulated at the check node. Subsequently, we can remove the known bit node and its outgoing edges from the graph. This procedure gives rise to a sequence of residual graphs. Moreover, when the sequence of residual graphs reaches the empty graph, this is a successful decoding.

Fig. 8 shows the process of the peeling decoder to the simple irregular LDPC code assuming that the received word is (0, ?, ?, 0, 1). The top left-most picture shows the initial graph and the received word. The following two pictures are part of the initialization: (i) known variable nodes send their values along all outgoing edges; (ii) these values are accumulated at the check nodes (the red box indicates an accumulated value of 1 and the orange one indicates an accumulated value of 0) and all known bit nodes and their connected edges are removed. After the initialization each decoding step consists of the following: (i) choose a check node of residual-degree one uniformly at random and forward the accumulated value to the connected bit node whose value is now determined; (ii) delete the chosen check node and its connected edge and forward the value of the newly determined bit node to all its remaining neighboring check nodes; (iii) accumulate the forwarded values at the check nodes and delete the bit node and its connected edges.

The example here has only a single check node of residual-degree one at each step. After two decoding steps, only empty graph left here, which is the residual graph we mentioned before (i.e., the decoder has succeeded in determining the codeword).

J. Tightness of the Upper Bound (Lower Bound)

For the BEC, the LDPC code can be decoded with a peeling decoder until the decoder gets stuck. If one analyzes this process carefully, one can compute the d.d. of the

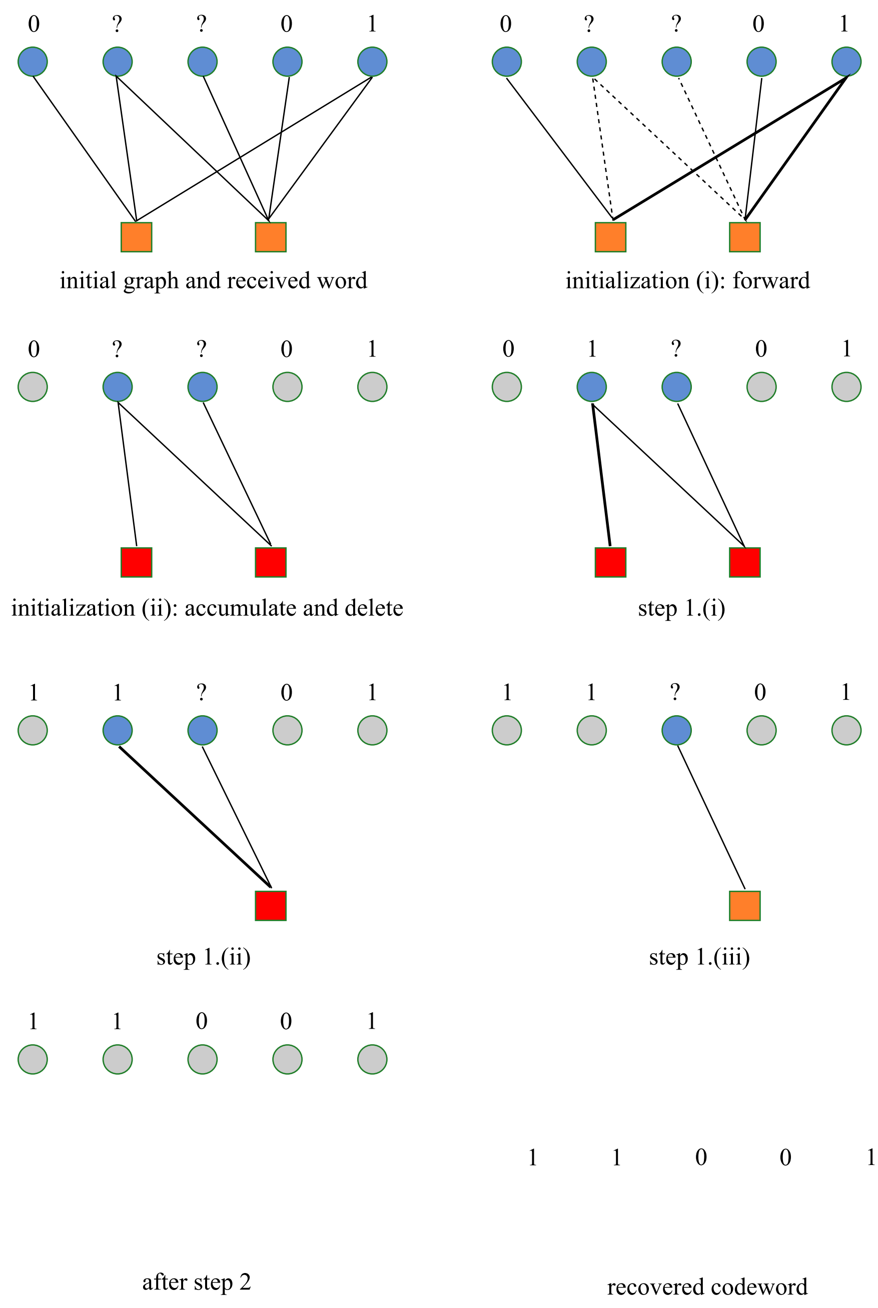

Fig. 8. Peeling decoder applied to simple irregular LDPC code with the received word (0, ?, ?, 0, 1). Through two decoding steps the peeling decoder has successfully recovered the codeword as (0, 1, 0, 0, 1).

residual graph and apply the counting argument of [4], [5] to prove the tightness of the MAP threshold.

Here are the steps of this concept:

Step 1: The LDPC code can be decoded with a peeling decoder until the decoder gets stuck (i.e., stopping set).

Step 2: The residual graph can be characterized in terms of the channel erasure probability ϵ (see *Remark* 5).

Step 3: Determine the ϵ where a MAP decoder applied to the residual graph succeeds with high probability (see *Remark* 6).

Remark 5. [16] Let ϵ denote the erasure probability of the channel and x denote the corresponding fixed point of density evolution, i.e., the largest solution of the equation $\epsilon\lambda\left(1-\rho\left(1-x\right)\right)=x$. Further, define $y=1-\rho\left(1-x\right)$. Assume we apply the peeling decoder. Then, at the fixed point, the expected degree distribution of the residual graph, call it $(\tilde{L}(z),\, \tilde{R}(z))$, has the form

$$\tilde{L}\left(z\right)\triangleq\frac{L\left(zy\right)}{L\left(y\right)},$$

$$\tilde{R}\left(z\right)\triangleq\frac{R\left(1-x+zx\right)-R\left(1-x\right)-zxR'\left(1-x\right)}{1-R\left(1-x\right)-xR'\left(1-x\right)}.$$

Remark 6. [16, p. 78 Lemma 3.22] Use the $\tilde{L}\left(z\right)$ and $\tilde{R}\left(z\right)$ we get from Step 2 and put in (3.23) (3.24) in [16]. See whether function $\Psi\left(y\right)$ has a unique maximum or not (i.e, reach the zero point at $y=1$). If it is, based on the Lemma here, it means that all other equations are linearly independent with high probability. (i.e, unique solution for the matrix with high probability). As a result, decoding will succeed with high probability.

Remark 7. (For step 2 and 3) First, we can choose ϵ^* is equal to the upper bound on ϵ^{MAP} and go through the step 2 and 3. Check whether function $\Psi(y)$ has a unique maximum. If it is, there is an unique solution for the matrix with high probability. This ϵ^* gives a lower bound on the MAP threshold. However, if function $\Psi(y)$ does not have a unique maximum (i.e., it does not reach the zero point at $y = 1$), there is a need to lower a little bit value of ϵ^*. Then, repeat step 2 and 3. Repeat the process until function $\Psi(y)$ has a unique maximum. Therefore, this method used to show the upper bound is tight for regular LDPC codes.

CHAPTER III

MAP THRESHOLD BOUNDS FOR IRA AND ARA CODES

A. Background on IRA and ARA Codes

IRA and ARA codes can be viewed as subclasses of LDPC codes that have natural linear-time encoding algorithms [9], [10]. Using iterative sum-product decoding, they can also be decoded with a per-iteration complexity that is linear in the block length. From an encoding point of view, it is natural to view IRA and ARA codes as interleaved serially concatenated codes [11]. From a decoding point of view, they are easily seen to be sparse-graph codes compatible with belief propagation decoding. There are a few slightly different definitions of ARA ensemble, and this thesis uses the ensemble and DE equations defined in [11].

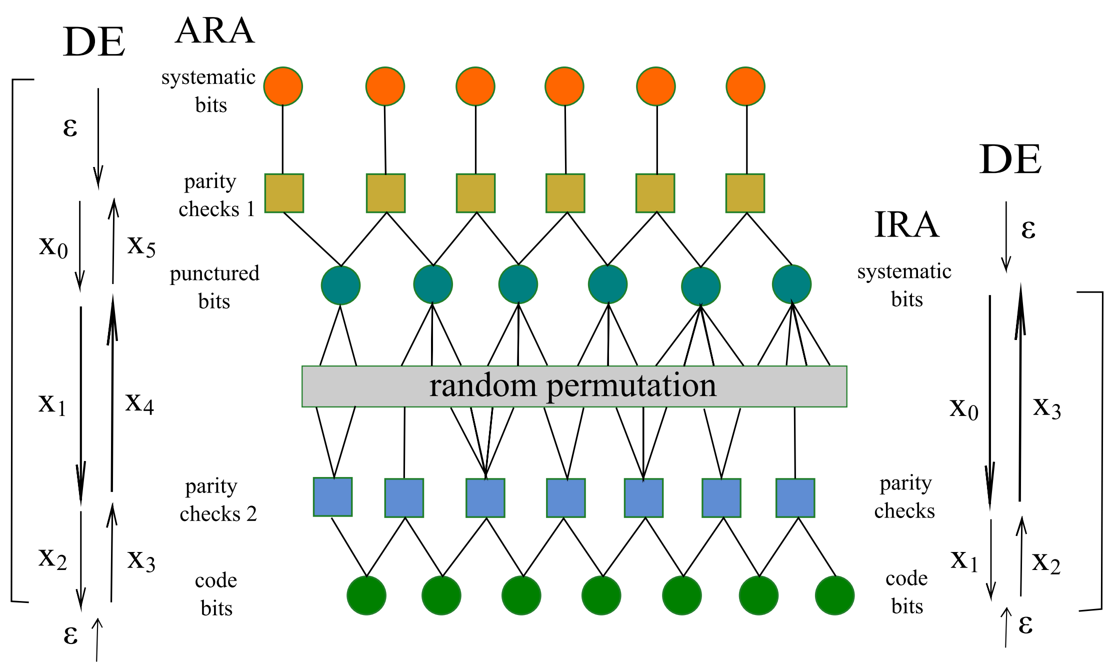

Fig. 9. Gallager-Tanner-Wiberg graph for ARA and IRA codes

B. MAP Threshold Bounds for Systematic IRA Codes

1. Density Evolution and Fixed Point Analysis of Iterative Decoding for IRA Codes

Since IRA codes can be viewed as LDPC codes with an accumulate structure attached to the check nodes (see Fig. 9), they can also be defined by the degree distribution of the bipartite graph between the systematic bits and the parity checks. Similar to Chapter II, we have

$$\lambda(x) = \sum_{i=1}^{\infty} \lambda_i x^{i-1} \quad and \quad \rho(x) = \sum_{i=1}^{\infty} \rho_i x^{i-1},$$

where λ_i (or ρ_i) represent the fraction of edges attached to a bit node (or check node) of degree i

In addition, from a node perspective, the degree distribution of bit and check nodes ($L(x)$ and $R(x)$) is also defined as same as those in Chapter II.

A random code is chosen from the ensemble (using a random permutation between bit and check nodes) and a random codeword is transmitted over a BEC with erasure probability ϵ. The asymptotic performance of the iterative message-passing decoder (as the block length of the code goes to infinity) is analyzed by tracking the average fraction of erasure messages which are passed the the graph of Fig. 9 during the l^{th} iteration. Also, the main assumption of DE is that the message passed on the edges of the Tanner graph are statistically independent. This assumption is justified by the fact that, for randomly chosen codes, the fraction of bits involved in finite-length cycles vanishes as the block length tends to infinity.

A single decoding iteration for IRA codes consists of four small steps which are performed on the Tanner graph of Fig. 9. Messages are first passed downward from the "systematic bit" nodes through each layer to the "code bit" nodes. Then, messages are passed back upwards from the "code bit" nodes through each layer to the "systematic bit" nodes. Let l designate the iteration number. Referring to Fig. 9, let $x_0^{(l)}$ and $x_3^{(l)}$ designate the probabilities of an erasure message from the "systematic bit" nodes to the "parity check" nodes and vice-versa, let $x_1^{(l)}$ and $x_2^{(l)}$ designate the probabilities of an erasure message from the "parity check" nodes to the "code bit" nodes and vice-versa.

From the Tanner graph of IRA codes in Fig. 9, we see that an outgoing message from a "systematic bit" node to a "parity check" node is an erasure if and only if all the incoming messages passed through the other edges connected to this bit are erasures. Using the statistical independence assumption, this yields the recursive equation

$$x_0^{(l)} = \epsilon \lambda \left(x_3^{(l)} \right)$$

It is also clear from Fig. 9 that an outgoing message from a "parity check" node to a "code bit" node is an erasure if either the incoming message through the other edge (which connects a "code bit" node to the same "parity check" node) is an erasure or the messages received from other edges (which connect a "systematic bit" node to the same "parity check" node) is an erasure. The update rule of the iterative message-passing decoder on the BEC therefore implies that

$$x_1^{(l)} = 1 - \left(1 - x_2^{(l-1)} \right) R \left(1 - x_0^{(l-1)} \right).$$

For any fixed number of decoding iterations l, the DE equations give (almost surely as $n \to \infty$) the erasure rate of the internal messages passed by the BP decoder for a random code and channel erasure pattern. In [9], with a similar process mentioned above, any fixed ϵ, the DE equations of the iterative message-passing decoder are given by

$$
\begin{aligned}
x_0^{(l)} &= \epsilon\lambda\left(x_3^{(l)}\right) \\
x_1^{(l)} &= 1-\left(1-x_2^{(l-1)}\right)R\left(1-x_0^{(l-1)}\right) \\
x_2^{(l)} &= \epsilon x_1^{(l)} \\
x_3^{(l)} &= 1-\left(1-x_2^{(l)}\right)^2\rho\left(1-x_0^{(l-1)}\right),
\end{aligned}
$$

where ϵ is the channel erasure probability and $x_i^{(l)}$ tracks the average fraction of erasure messages for edge type-i and iteration l.

The rate r_{IRA} of a systematic IRA code given by the degree distribution can be written as

$$
r_{IRA} = \left(1+\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}\right)^{-1}.
$$

2. BP EXIT Function and Bounds on the MAP Threshold

Lemma 1. The asymptotic BP EXIT function of the IRA code ensemble is given by

$$
h^{BP-IRA}(\epsilon) = r_{IRA}\,L\left(x_3\right) + \left(1-r_{IRA}\right)x_1^2,
$$

where x_0, x_3 are given by the $l \to \infty$ DE fixed point for that ϵ.

Proof. IRA codes have multiple types of bits in the GTW graph. The $n(1-r)$ parity bits have an average extrinsic erasure probability of $(x_1^{(l)})^2$ after l iterations. Likewise, the nrL_d information bits of degree-d have an average extrinsic erasure probability of

$(x_3^{(l)})^d$ after l iterations. Therefore, we can write the large-iteration long-block limit of the IRA code EXIT function as

$$\begin{aligned} h^{BP-IRA}(\epsilon) &= \lim_{l\to\infty}\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} h_i^{BP-IRA}(\epsilon) \\ &\overset{a.s.}{=} \lim_{l\to\infty}\left[r_{IRA}\sum_{d=1}^{\infty} L_d\left(x_3^{(l)}\right)^d + (1-r_{IRA})\left(x_1^{(l)}\right)^2\right] \\ &= r_{IRA}\, L(x_3) + (1-r_{IRA})\,(x_1)^2 \,. \end{aligned}$$

□

Accordingly, we plot the BP EXIT function of the IRA code and integrate backwards from the right end of the curve where $\epsilon = 1$. The integration process stops at ϵ^* when $\int_{\epsilon^*}^{1} h^{BP}(\epsilon)\, d\epsilon = r_{IRA}$. This gives the upper bound $\epsilon^{MAP} \le \epsilon^*$ for the IRA code ensemble (see Fig. 10). Moreover, Table. I shows the comparison of the thresholds for various IRA codes ensembles.

C. MAP Threshold Bounds for ARA Codes

1. Density Evolution of Systematic ARA Ensembles

An irregular ensemble of ARA codes can also be defined by its degree distribution pair $\lambda(x), \rho(x)$ (from an edge perspective), and $L(x), R(x)$ (from a node perspective).

Pfister and Sason [11] consider the asymptotic analysis of ensembles of ARA codes under the assumption that a random codeword is transmitted over a BEC with erasure probability ϵ.

Similar to the algorithm we mentioned earlier, a single decoding iteration for ARA codes consists of six small steps which are performed on the Tanner graph of Fig. 9. Messages are first passed downward from the "systematic bit" nodes through

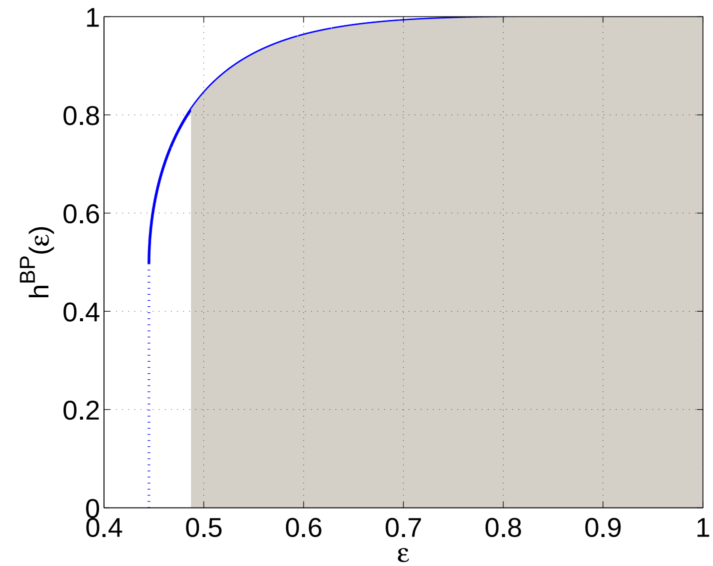

Fig. 10. The BP EXIT function $h^{BP}(\epsilon)$ of a (4, 4)-regular IRA code (i.e., $\lambda(x) = \rho(x) = x^3$) on the erasure channel. The left boundary of the shaded area is the upper bound on ϵ^{MAP}, and the area of the shaded portion under the curve equals the code rate of $\frac{1}{2}$. This gives $\epsilon^{BP} = 0.4451$ and $\epsilon^{MAP} \leq 0.4872$. (Note: $\epsilon^{Shannon} = 0.5$)

each layer to the "code bit" nodes. Then, messages are passed back upwards from the "code bit" nodes through each layer to the "systematic bit" nodes. Let l designate the iteration number. Referring to Fig. 9, let $x_0^{(l)}$ and $x_5^{(l)}$ designate the probabilities of an erasure message from the "parity-check 1" nodes to the "punctured bit" nodes and vice-versa, let $x_1^{(l)}$ and $x_4^{(l)}$ designate the probabilities of an erasure message from the "punctured bit" nodes to the "parity-check 2" nodes and vice-versa, and finally, let $x_2^{(l)}$ and $x_3^{(l)}$ designate the probabilities of an erasure message from the "parity-check 2" nodes to the "code bit" nodes and vice-versa.

From the Tanner graph of ARA codes in Fig. 9, we see that an outgoing message from a "parity-check 1" node to a "punctured bit" node is an erasure if either the incoming message through the other edge (which connects a "punctured bit" node

Table I. Comparison of the thresholds for various IRA codes ensembles.

$\lambda(x)$	$\rho(x)$	ϵ^{BP}	ϵ^{MAP} Upper Bound	$\epsilon^{Shannon}$	rate
x^3	x^3	0.4451	0.4872	0.5	0.5
x^2	x^2	0.4448	0.4651	0.5	0.5
x^5	x^3	0.5186	0.5967	0.6	0.4
x^6	x^8	0.3311	0.4366	0.4375	0.5625
x^2	$\frac{2x+1}{3}$	0.6043	0.6227	0.6667	0.3333
$\frac{7x^7+3x^2}{10}$	x^5	0.4182	0.4666	0.4706	0.5294
x^6	$\frac{806x^9+22x^2+172x}{1000}$	0.4309	0.5317	0.5490	0.4510
$\frac{8x^4+x^3+x}{10}$	$\frac{3x^4+2x^2+2}{7}$	0.5999	0.6308	0.6651	0.3349

to the same "parity-check 1" node) is an erasure or the message received from the BEC for the systematic bit (which is connected to the same "parity-check 1" node) is an erasure. Using the statistical independence assumption, this yields the recursive equation

$$x_0^{(l)} = 1 - (1 - \epsilon)\left(1 - x_5^{(l-1)}\right)$$

It is also clear from Fig. 9 that an outgoing message from a "punctured bit" node to a "parity-check 2" node is an erasure if and only if all the incoming messages passed through the other edges connected to this bit are erasures. The update rule of the iterative message-passing decoder on the BEC therefore implies that

$$x_1^{(l)} = \left(x_0^{(l)}\right)^2 \lambda\left(x_4^{(l-1)}\right)$$

For any fixed number of decoding iterations l, the DE equations give (almost surely as $n \to \infty$) the erasure rate of the internal messages passed by the BP decoder for a random code and channel erasure pattern. In [11], with a similar process mentioned above, any fixed ϵ, the DE equations for the BEC can be computed in closed form. From Fig. 9, we see that

$$\begin{aligned}
x_0^{(l)} &= 1 - \left(1 - x_5^{(l-1)}\right)(1-\epsilon) \\
x_1^{(l)} &= \left(x_0^{(l)}\right)^2 \lambda \left(x_4^{(l-1)}\right) \\
x_2^{(l)} &= 1 - R\left(1 - x_1^{(l)}\right)\left(1 - x_3^{(l-1)}\right) \\
x_3^{(l)} &= \epsilon x_2^{(l)} \\
x_4^{(l)} &= 1 - \left(1 - x_3^{(l)}\right)^2 \rho \left(1 - x_1^{(l)}\right) \\
x_5^{(l)} &= x_0^{(l)} L\left(x_4^{(l)}\right),
\end{aligned}$$

where ϵ is the channel erasure probability and $x_i^{(l)}$ tracks the average fraction of erasure messages for edge type-i and iteration l.

The rate r_{ARA} of a systematic ARA code is computed by expressing the block length n as the sum of k systematic bits and $kL'(1)/R'(1)$ parity bits which then yields

$$r_{ARA} = \frac{1}{1 + \frac{L'(1)}{R'(1)}}.$$

2. BP EXIT Function and Bounds on the MAP Threshold

Lemma 2. The asymptotic BP EXIT function of the ARA code ensemble is given by

$$h^{BP-ARA}(\epsilon) = r_{ARA}\left[1 - (1 - x_5)^2\right] + (1 - r_{ARA})\, x_2^2,$$

where x_5, x_2 are given by the $l \to \infty$ DE fixed point for that ϵ.

Proof. ARA codes have two classes of bits are transmitted across the channel. The nr systematic bits have an average extrinsic erasure probability of $1 - (1 - x_5^{(l)})^2$ after l iterations. Likewise, the $n\,(1-r)$ code bits have an average extrinsic erasure probability of $(x_2^{(l)})^2$, after l iterations. Thus, we can write the large-iteration long-block limit of the ARA code EXIT function as

$$\begin{aligned} h^{BP-ARA}(\epsilon) &= \lim_{l\to\infty}\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} h_i^{BP-ARA}(\epsilon) \\ &\overset{a.s.}{=} \lim_{l\to\infty}\left[r_{ARA}\left(1-\left(1-x_5^{(l)}\right)^2\right)+(1-r_{ARA})\left(x_2^{(l)}\right)^2\right] \\ &= r_{ARA}\left[1-(1-x_5)^2\right]+(1-r_{ARA})\,x_2^2. \end{aligned}$$

□

The same integration process, that was used for LDPC and IRA codes, is used to calculate the upper bound on ϵ^{MAP} for ARA codes (see Fig. 11). Moreover, Table. II shows the comparison of the thresholds for various ARA codes ensembles.

Table II. Comparison of the thresholds for various ARA codes ensembles.

$\lambda(x)$	$\rho(x)$	ϵ^{BP}	$\epsilon^{MAP} Upper Bound$	$\epsilon^{Shannon}$	rate
x^2	$\frac{2x+1}{3}$	0.6412	0.6593	0.6667	0.3333
x^2	$\frac{x^2+x+1}{3}$	0.6186	0.6383	0.6471	0.3529
$\frac{4x^2+x}{5}$	$\frac{2x+1}{3}$	0.6248	0.6352	0.6452	0.3548
$\frac{x^2+x}{2}$	$\frac{3x^3+2x^2+x+1}{7}$	0.4780	0.4922	0.5000	0.5000
x	$\frac{4x^3+1x^2+x+1}{7}$	0.4316	0.4365	0.4474	0.5526
$\frac{3x^2+x+1}{5}$	$\frac{6x^3+x^2+x+1}{9}$	0.4153	0.4169	0.4255	0.5745

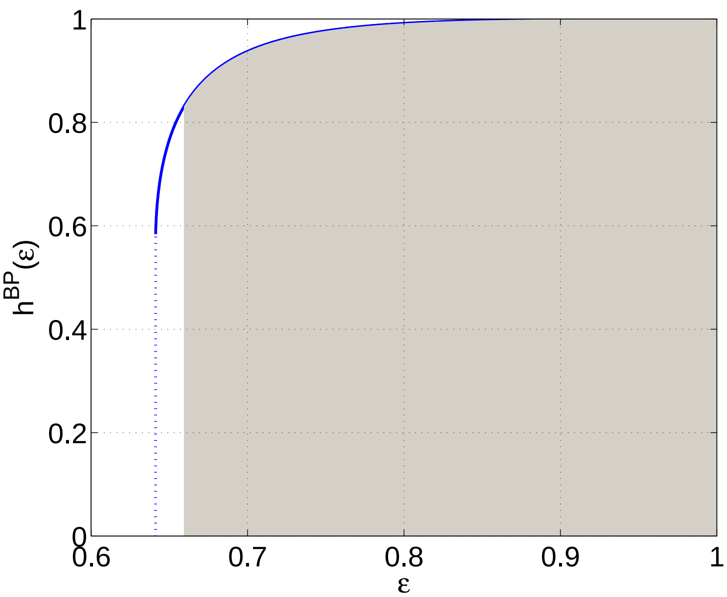

Fig. 11. The BP EXIT function $h^{BP}(\epsilon)$ of the ARA code with $\lambda(x) = x^2$, $\rho(x) = \frac{2}{3}x + \frac{1}{3}$ on the erasure channel. The left boundary of the shaded area is the upper bound on ϵ^{MAP}, and the area of the shaded portion under the curve equals the code rate of $\frac{1}{3}$. This gives $\epsilon^{BP} = 0.6412$ and $\epsilon^{MAP} \leq 0.6593$. (Note: $\epsilon^{Shannon} = 0.6666$)

D. Tightness of the Upper Bound on IRA and ARA codes (Lower Bound)

For the BEC, graph reduction can be used to reduce any IRA or ARA code into an LDPC code [11]. After this reduction, the LDPC code can be decoded with a peeling decoder until the decoder gets stuck. If one analyzes this process carefully, one can compute the d.d. of residual graph and apply the the counting argument of [4], [5] to (possibly) prove the tightness of the MAP threshold.

Remark 8. Pfister and Sason [11] show the new d.d. of the bit and check nodes after the graph reduction is given, respectively, by

$$\tilde{R}(x) = \sum_{k=0}^{\infty} \epsilon^k (1-\epsilon) R(x)^{k+1} = \frac{(1-\epsilon) R(x)}{1-\epsilon R(x)},$$

and

$$\tilde{L}\left(x\right) = \sum_{k=0}^{\infty} \left(1-\epsilon\right)^{k} \epsilon L\left(x\right)^{k+1} = \frac{\epsilon L\left(x\right)}{1-\left(1-\epsilon\right) L\left(x\right)}.$$

After this reduction, we can go through the steps described in Section J of Chapter II. Therefore, the tightness of the MAP threshold on IRA and ARA codes can (possibly) be proved.

CHAPTER IV

MAP THRESHOLD BOUNDS FOR JOINT DECODING

A. Background and System Description

Pfister and Siegel in [14] consider the achievable rate of joint iterative decoding of LDPC codes and channels with memory. Here we use same system model and consider instead the MAP decoding threshold. The block diagram of the system is shown in Fig. 12. It is a relatively standard setup for the joint iterative decoding of an LDPC code and a channel with memory. Equiprobable information bits, $U_1^k \in \{0,1\}^k$, are encoded into an LDPC codeword, $X_1^n \in \{0,1\}^n$, which is observed through the dicode erasure channel (DEC) as the output vector, $Y_1^n \in \{-1,0,1,?\}$. The decoder consists of the channel APP detector an LDPC decoder which pass messages back and forth. In the first half of decoding iteration i, the channel detector decodes Y_1^n using the *a priori* information, $W_1^n \in \{0,1,?\}$, from the LDPC code. In the second half of decoding iteration i, one LDPC decoding iteration is completed using internal edge messages from the previous LDPC iteration and the output of the channel detector. A random scrambling sequence is added to the codeword before transmission and removed before LDPC decoding; this is very similar to using a random coset of the LDPC code. Fig. 13 shows the GTW graph of the joint iterative decoder.

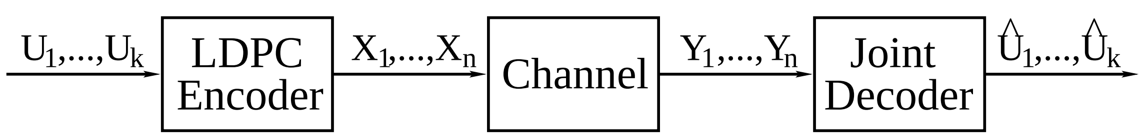

Fig. 12. Block diagram of the system.

1. The Dicode Erasure Channel

The dicode erasure channel (DEC) is a binary input channel based on the $1-D$ linear intersymbol-interference (ISI) dicode channel. The output of the $1-D$ channel with binary inputs (e.g., +1, 0, -1) is erased with probability ϵ and transmitted perfectly with probability $1-\epsilon$. More information about the DEC can be found in [20], [14].

The simplicity of the DEC allows the BCJR algorithm for the channel to be analyzed in closed form. The method is similar to the exact analysis of turbo codes on the BEC [5]. The EXIT function for the DEC is computed in [14] and if the outputs are erased with probability ϵ, then the EXIT function of the channel detector is given by

$$f(x;\epsilon) = \frac{4\epsilon^2}{\left(2 - x(1-\epsilon)\right)^2}.$$

The capacity of the DEC for independent equiprobable inputs can be computed by analyzing only the forward recursion of the BCJR algorithm [20] and is given by

$$C_{i.u.d.}(\epsilon) = 1 - \frac{2\epsilon^2}{1+\epsilon}.$$

B. Density Evolution for Joint Decoding

The closed form analysis of this system is based on the fact that all the messages passed in decoding graph are erasure messages. This allows DE of the joint iterative decoder to be represented by a single parameter recursion. Let $f(x;\epsilon)$ be a function which maps the erasure probability, x, of the *a priori* LLR distribution to the erasure probability at the output of the channel detector for a channel erasure probability of ϵ. Following [14], we refer to $f(x;\epsilon)$ as the *extrinsic information transfer* (EXIT)

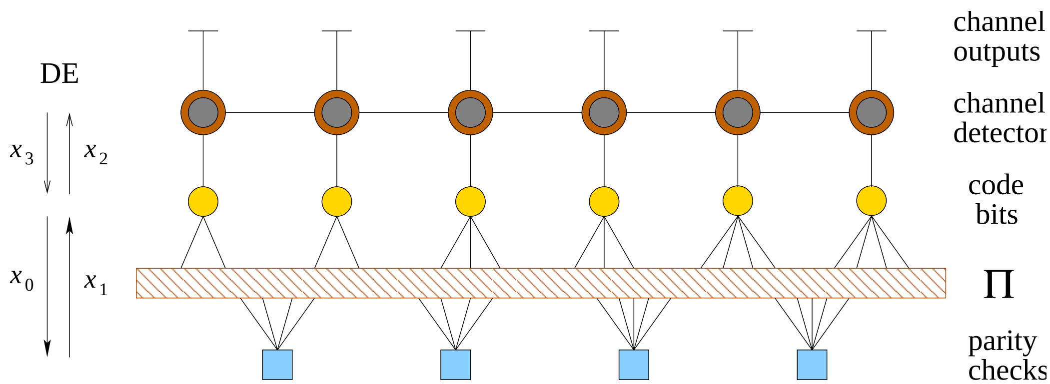

Fig. 13. Gallager-Tanner-Wiberg graph of the joint iterative decoder.

function of the channel.

The joint decoding graph is shown in Fig. 13 and message-passing schedule and variables are shown on the left. Let $x_0^{(l)}$, $x_1^{(l)}$, $x_2^{(l)}$, and $x_3^{(l)}$ denote the erasure rate of messages passed during iteration l. The update equations are as follows

$$x_0^{(l+1)} = x_3^{(l)} \lambda \left(x_1^{(l)} \right)$$
$$x_1^{(l+1)} = 1 - \rho \left(1 - x_0^{(l+1)} \right)$$
$$x_2^{(l+1)} = L \left(x_1^{(l+1)} \right)$$
$$x_3^{(l+1)} = f \left(x_2^{(l+1)}; \epsilon \right).$$

The first two equations simply describe LDPC decoding when the channel erasure parameter is $x_3^{(l)}$ instead of the fixed constant ϵ. The third equation describe the message passing from the code to the channel detector. The fourth equation takes the channel detector bits into account and simply maps side information from the code through the EXIT function $f(x; \epsilon)$.

C. The EXIT Area Theorem for Joint Decoding

In this section, we consider the MAP EXIT function of the entire joint decoder. Consider any FS channel with deterministic ISI that is observed through an erasure channel. In this case, the output sequence Y_1^n consists of independently erased observations (with probability ϵ) of a deterministic sequence Z_1^n that, given the initial state S_1, is in one-to-one correspondence with the input sequence X_1^n.

Definition 3. Then, the *joint decoding MAP EXIT function* is defined to be

$$h^{MAP-JD}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} H\left(Z_i | Y_1^n(\epsilon) \backslash Y_i(\epsilon), S_1\right).$$

Definition 4. Then, the *joint decoding BP EXIT function* is defined to be

$$h^{BP-JD}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} h_i^{BP-JD}(\epsilon),$$

where $h_i^{BP-JD}(\epsilon)$ is the entropy of the iterative decoding estimate of Z_i from $Y_1^n \backslash Y_i$ and S_1. The iterative decoding estimate of this output symbol is given by the symbol's extrinsic message in the joint decoder after l iterations of decoding.

Corollary 1. Let X_1^n be chosen according to $p_{X_1^n}(x_1^n)$ and Y_1^n be the result of transmitting X_1^n over the above FS ISI channel. To emphasize that Y_1^n depends on the channel parameter δ write $Y_1^n(\delta)$. Then

$$\frac{1}{n} H\left(X_1^n | Y_1^n(\delta)\right) = \int_0^{\delta} h^{MAP-JD}(\epsilon)\, d\epsilon.$$

Proof. The proof is a slight modification of the approach taken in [4], [5]. The one-to-one correspondence between X_1^n and Z_1^n (given S_1) implies that

$$\begin{aligned} H\left(X_1^n|Y_1^n,S_1\right) &= H\left(Z_1^n|Y_1^n,S_1\right) \\ &= H\left(Z_i|Y_1^n,S_1\right) + H\left(Z_1^n\backslash Z_i|Y_1^n,Z_i,S_1\right). \end{aligned}$$

Since Y_i is a noisy observation of Z_i, we find that $Z_1^n\backslash Z_i \rightarrow Z_i \rightarrow Y_i$ forms a Markov chain. Moreover, if each channel mapping $Z_i \rightarrow Y_i$ depends on a different parameter ϵ_i, then we can write

$$\frac{d}{d\epsilon_i} H\left(X_1^n|Y_1^n,S_1\right) = \frac{d}{d\epsilon_i} H\left(Z_i|Y_1^n,S_1\right)$$

because $H\left(Z_1^n\backslash Z_i|Y_1^n,Z_i,S_1\right)$ is independent of ϵ_i. If we assume also that Y_i is either an erasure (with probability ϵ_i) or deterministic function $Y_i = Z_i$ (with probability $1-\epsilon_i$), then we can write

$$\begin{aligned} H\left(Z_i|Y_1^n,S_1\right) = {} & \epsilon_i H\left(Z_i|Y_1^n \setminus Y_i,S_1\right) \\ & + \left(1-\epsilon_i\right) H\left(Z_i|Y_1^n \setminus Y_i,Z_i,S_1\right) \end{aligned}$$

Using the derivative method, this gives

$$\begin{aligned} \frac{d}{d\epsilon} H\left(X_1^n|Y_1^n\right) &= \sum_{i=1}^{n} \frac{d}{d\epsilon_i} H\left(Z_i|Y_1^n,S_1\right) \\ &= \sum_{i=1}^{n} H\left(Z_i|Y_1^n \setminus Y_i,S_1\right). \end{aligned}$$

□

D. BP EXIT Function and Bounds on the MAP Threshold

The joint decoding BP EXIT function simply can be computed by analyzing the BCJR decoding algorithm for the DEC. Since the channel has only two states and the channel inputs satisfy $\Pr(X_i = 0) = \Pr(X_i = 1) = \frac{1}{2}$, the forward and backward recursion vectors (which can have infinite support) effectively take only two values; the state is either known (denoted $\mathcal{K}$) or unknown (denoted $\mathcal{U}$) [14].

To compute the BP EXIT function, we can simply analyze the output stage of the BCJR algorithm and compute the entropy of Z_i given the current messages. Notice that, at any point in the trellis, there are four distinct possibilities for forward/backward recursion (α/β) state knowledge: $(\mathcal{K}/\mathcal{K})$, $(\mathcal{K}/\mathcal{U})$, $(\mathcal{U}/\mathcal{K})$, and $(\mathcal{U}/\mathcal{U})$. Let ϵ be the channel erasure rate and $\delta = L(x_1)$ be the *a priori* erasure rate from the LDPC code, and define $H_{AB} = H(Z_i|\alpha_i \in A, \beta_{i+1} \in B)$.

Accordingly, first, for the state $(\mathcal{K}/\mathcal{K})$, entropy of the channel output is 0 ($H_{\mathcal{K}\mathcal{K}} = 0$). Second, for the state $(\mathcal{K}/\mathcal{U})$, entropy of the channel output is 1 if *a priori* symbol $W = ?$; otherwise, entropy of the channel output is 0. Third, for the state $(\mathcal{U}/\mathcal{K})$, entropy of the channel is 1. Finally, for the last state $(\mathcal{U}/\mathcal{U})$, entropy of the channel output is 3/2 if *a priori* symbol $W = ?$; otherwise, entropy of the channel output is 1. Then, the entropy conditioned on the trellis edge is given by

$$\begin{aligned} H_{\mathcal{K}\mathcal{K}} &= 0 \qquad H_{\mathcal{K}\mathcal{U}} = \delta \\ H_{\mathcal{U}\mathcal{K}} &= 1 \qquad H_{\mathcal{U}\mathcal{U}} = \frac{3}{2}\delta + (1-\delta). \end{aligned}$$

From [14], the steady state probability that the forward/backward recursion has no state knowledge is

$$\Pr\left(\alpha \in \mathcal{U}\right) = \frac{2\epsilon\delta}{2-\delta\left(1+\epsilon\right)+2\epsilon\delta}$$
$$\Pr\left(\beta \in \mathcal{U}\right) = \frac{2\epsilon}{\left(1-\epsilon\right)\left(2-\delta\right)+2\epsilon}.$$

Therefore, the asymptotic BP EXIT function of the system can be written as

$$\begin{aligned} h^{BP-JD}\left(\epsilon\right) =& \Pr\left(\alpha \in \mathcal{K}, \beta \in \mathcal{U}\right) \cdot \delta + \Pr\left(\alpha \in \mathcal{U}, \beta \in \mathcal{K}\right) \\ &+ \Pr\left(\alpha \in \mathcal{U}, \beta \in \mathcal{U}\right) \cdot \left[\frac{3}{2}\delta + \left(1-\delta\right)\right] \\ =& \frac{\delta\left(\left(\delta-2\right)^2 - 2\epsilon\left(\delta-2\right) - \delta\epsilon^2\left(\delta-4\right)\right)}{\left(2-\delta\left(1-\epsilon\right)\right)^2}, \end{aligned}$$

where $\delta = L(x_1)$ and x_1 is given by the $l \to \infty$ DE fixed point for that ϵ.

Fig. 14 shows the BP EXIT function $h^{BP}\left(\epsilon\right)$, and area under the BP EXIT function curve equals the code rate $\frac{1}{2}$. The left boundary of the integration area is the upper bound on ϵ^{MAP} of the joint iterative decoder for a (3,6)-regular LDPC code and the DEC channel. Moreover, Table. III shows the comparison of the thresholds of the joint iterative decoder for various LDPC codes ensembles.

Table III. Comparison of the thresholds of the joint iterative decoder for various LDPC codes ensembles.

$\lambda(x)$	$\rho(x)$	ϵ^{BP}	ϵ^{MAP}	rate
x^2	x^5	0.5689	0.6430	0.5
x	x^6	0.4332	0.4662	0.7143
x^2	$\frac{x^3+x^2}{2}$	0.8158	0.9151	0.1250

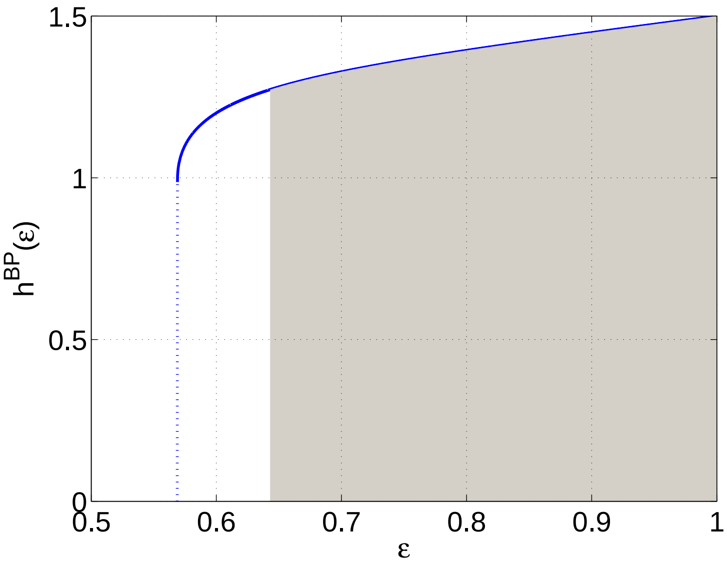

Fig. 14. The BP EXIT function $h^{BP}(\epsilon)$ of the joint iterative decoder for a (3,6)-regular LDPC code and the DEC channel. The left boundary of the shaded area is the upper bound on ϵ^{MAP}, and the area of the shaded portion under the curve equals the code rate of $\frac{1}{2}$. This gives $\epsilon^{BP} = 0.5689$ and $\epsilon^{MAP} \leq 0.6430$.

E. Tightness of the Upper Bound (Lower Bound)

For the BEC, finding the lower bound seems difficult for the joint decoder since joint decoder includes both the DEC channel and LDPC codes. There is no clear method to achieve the lower bound on it. While there may be an analogous graph reduction method to prove the tightness of this bound, we were unable to find it during our study. It seems that it is not as straightforward as the IRA and ARA case.

CHAPTER V

CONCLUSIONS

The direct computation of the MAP threshold ϵ^{MAP} is infeasible. Therefore, in this thesis, based on the techniques introduced by Masson, Montanari, and Urbanke in [4], [5], upper bounds on the MAP thresholds are computed for three iterative decoding systems: IRA codes, ARA codes, the joint decoding of LDPC codes, and channels with memory. The bound for joint decoding requires a slight generalization of the EXIT area theorem that is introduced within.

In addition, the difference between ϵ^{BP} and ϵ^{MAP} quantifies the loss due to iterative decoding. Also, the difference between ϵ^{MAP} and $\epsilon^{Shannon}$ of IRA and ARA codes ensemble shows the loss for that code. From Tab. I, and Tab. II, they clearly show the two gaps of many various codes.

Some open questions include the tightness of these bounds and the extensions to non-erasure channels. For tightness of these bounds, progress is made for IRA and ARA codes ensembles. However, it seems difficult for the joint decoder since joint decoder includes both the DEC channel and LDPC codes. There is no clear method to achieve the lower bound on it. For non-erasure channels, these bounds also have natural extensions by way of the generalized EXIT (GEXIT) functions introduced in [21]. Moreover, the lower bound is still open since there is no peeling decoder for non-erasure channels.

REFERENCES

[1] Claude E. Shannon, "A mathematical theory of communication," *The Bell Syst. Techn. J.*, vol. 27, pp. 379–423, 623–656, July / Oct. 1948.

[2] Robert G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, M.I.T., Cambridge, MA, USA, 1960.

[3] Claude Berrou, Alain Glavieux, and Punya Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, IEEE, vol. 2, pp. 1064–1070.

[4] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell's construction: The hidden bridge between iterative and maximum a posteriori decoding," *Arxiv preprint cs.IT/0506083v1*, June 2005.

[5] Cyril Méasson, "Conservation laws for coding," Ph.D. dissertation, Swiss Federal Institute of Technology, Lausanne, 2006.

[6] Stephan ten Brink, "Convergence of iterative decoding," *Electronic Letters*, vol. 35, no. 10, pp. 806–808, May 1999.

[7] Thomas J. Richardson, M. Amin Shokrollahi, and Rüdiger L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[8] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2674, Nov. 2004.

[9] Hui Jin, Aamod Khandekar, and Robert J. McEliece, "Irregular repeat-accumulate codes," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, Sept. 2000.

[10] A. Abbasfar, D. Divsalar, and Y. Kung, "Accumulate repeat accumulate codes," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, June 2004, p. 505.

[11] H. D. Pfister and I. Sason, "Accumulate–repeat–accumulate codes: Capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity," *IEEE Trans. Inform. Theory*, vol. 53, no. 6, pp. 2088–2115, June 2007.

[12] Catherine Douillard, Michel Jézéquel, Claude Berrou, Annie Picart, Pierre Didier, and Alain Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *Eur. Trans. Telecom.*, vol. 6, no. 5, pp. 507–511, Sept. – Oct. 1995.

[13] Brian M. Kurkoski, Paul H. Siegel, and Jack Keil Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1410–1422, June 2002.

[14] Henry D. Pfister and Paul H. Siegel, "Joint iterative decoding of LDPC codes for channels with memory and erasure noise," *IEEE J. Select. Areas Commun.*, vol. 26, no. 2, pp. 320–337, Feb. 2008.

[15] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell's construction: the hidden bridge between maximum-likelihood and iterative decoding," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, June 2004, p. 225.

[16] Thomas J. Richardson and Rüdiger L. Urbanke, *Modern Coding Theory*, Cambridge University Press, New York, NY, 2008.

[17] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[18] Thomas J. Richardson and Rüdiger L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[19] A. Montanari, "Tight bounds for LDPC and LDGM codes under MAP decoding," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3221–3246, 2005.

[20] Henry D. Pfister, "On the capacity of finite state channels and the analysis of convolutional accumulate-m codes," Ph.D. dissertation, University of California, San Diego, La Jolla, CA, USA, March 2003.

[21] C. Measson, A. Montanari, T. Richardson, and R. Urbanke, "The generalized area theorem and some of its consequences," *Arxiv preprint cs.IT/0511039*, 2005.

VITA

Chia-Wen Wang was born in Taipei, Taiwan. He received his Bachelor of Science degree in Engineering Science from National Cheng Kung University in June 2003, and Master of Science degree in Electrical and Computer Engineering from Texas A&M University in August 2008. His graduate thesis topic was "Bounds on the MAP Threshold of Iterative Decoding Systems with Erasure Noise". His primary interests include Channel Coding, Information Theory, Wireless Communications, and Signal Processing.

Mr. Wang can be reached at room 228A of Wisenbaker Engineering Research Center (WERC) of Texas A&M University. His email address is ryancwwang@gmail.com.