ON THE ANALYSIS OF SPATIALLY-COUPLED GLDPC CODES AND THE WEIGHTED MIN-SUM ALGORITHM

A Dissertation

by

YUNG-YIH JIAN

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Henry D. Pfister
Committee Members,	Krishna R. Narayanan
	Thomas Schlumprecht
	Srinivas Shakkottai
Head of Department,	Chanan Singh

August 2013

Major Subject: Electrical Engineering

ABSTRACT

This dissertation studies methods to achieve reliable communication over unreliable channels. Iterative decoding algorithms for low-density parity-check (LDPC) codes and generalized LDPC (GLDPC) codes are analyzed.

A new class of error-correcting codes to enhance the reliability of the communication for high-speed systems, such as optical communication systems, is proposed. The class of spatially-coupled GLDPC codes is studied, and a new iterative hard-decision decoding (HDD) algorithm for GLDPC codes is introduced. The main result is that the minimal redundancy allowed by Shannon's Channel Coding Theorem can be achieved by using the new iterative HDD algorithm with spatially-coupled GLDPC codes. A variety of low-density parity-check (LDPC) ensembles have now been observed to approach capacity with iterative decoding. However, all of them use soft (i.e., non-binary) messages and a posteriori probability (APP) decoding of their component codes. To the best of our knowledge, this is the first system that can approach the channel capacity using iterative HDD.

The optimality of a codeword returned by the weighted min-sum (WMS) algorithm, an iterative decoding algorithm which is widely used in practice, is studied as well. The attenuated max-product (AttMP) decoding and weighted min-sum (WMS) decoding for LDPC codes are analyzed. Applying the max-product (and belief-propagation) algorithms to loopy graphs is now quite popular for best assignment problems. This is largely due to their low computational complexity and impressive performance in practice. Still, there is no general understanding of the conditions required for convergence and/or the optimality of converged solutions. This work presents an analysis of both AttMP decoding and WMS decoding for LDPC codes

which guarantees convergence to a fixed point when a weight factor, β, is sufficiently small. It also shows that, if the fixed point satisfies some consistency conditions, then it must be both a linear-programming (LP) and maximum-likelihood (ML) decoding solution.

DEDICATION

To my family,

and in memory of my father-in-law (1924–2013)

ACKNOWLEDGEMENTS

I would like to thank all who have offered help throughout my journey to the Ph.D. degree, especially the following subset of people. Without their support and encouragement, it would not have been possible for me to finish this journey.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Henry Pfister. Since the first day of joining his group, his vast mathematical background and the ability of distilling the essence of each problem have continued to astound me. His valuable suggestions and encouragements in each discussion have been the most significant help in my research. His research philosophy, thinking big pictures but starting from small examples, has greatly benefited me in the past years. Besides the help on research, I also would like to thank for his generous help and support when I was looking for my future career. At the beginning of my doctoral study, I was blessed to be able to work with him and learn from him. Upon completing the degree, I hope I have also absorbed part of his great personality: kindness and generosity of heart, and optimism for life.

I am also grateful to Professor Krishna Narayanan for his generous help along the way. His great intuition and deep understanding in coding theory and information theory have made every discussion invaluable and inspirational. The knowledge I learned from his class, Advanced Channel Coding, has been one of the most important cornerstones of my research. I am also grateful to him for his help when I was looking for a job. I want to take this opportunity to express my sincere appreciation to Professor Srinivas Shakkottai and Professor Thomas Schlumprecht for serving on my dissertation committee. Their useful comments and generous support have greatly improved the quality of my research.

I want to thank Professor Jean-Francois Chamberland, Professor Tie Liu, Professor Shuguang Cui, Professor Scott Miller and Professor Ulisses Braga-Neto for their wonderful courses. I also want send my warm thanks to the ECE staff for their excellent administration support, particularly Tammy, Gayle, Claudia, Linda, and Anni. Their help made my stay at Texas A&M smooth and enjoyable.

Sincere thanks go to all my friends and colleagues in the lab. Fan Zhang for his constant warm encouragement in my early stage of study. Byung-Hak Kim for introducing me topics of machine learning. Phong Sy Nguyen for the mutual encouragement along the way and for sharing experiences of taking care of baby. Arvind Yedla for all the meetings and discussions in Sweet Eugene's that have been some of the most productive and enjoyable moments in my graduate life. Fatemeh Hamidi-Sepehr for interesting cultural discussions and her warm encouragement. Santhosh Kumar for interesting discussions on subjects ranging from analysis, probability to the life of Bernhard Riemann. Andrew Young for many fruitful discussions and all the coffee breaks in McDonald's. Janson Moore for his unique way of introducing me the American culture. Yu-Chih (Jerry) Huang for teaching me two-way relay channel. I also want to thank Yu-Pin Hsu, Wei-Yu Chen, Jae Won Yoo, Amir Salimi, Avinash Vem, Engin Tunali and Brett Hern for their company in the past years.

Last but not least, I express my warmest thanks to my parents and my parents-in-law for their constant love and support. In particular to my father-in-law, I would not have gone for study abroad without his encouragement. I especially want to thank my dear wife, Ching-Hua. She has supported me and believed in me throughout the years. More than that, she gave birth to our sweetest daughter, Alexis, even thought she was also busy in her Ph.D. study. I am deeply indebted to her for all her sacrifices and unconditional love.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Communication is one of the most important activities undertaken by human beings, and it has been a challenge throughout history to convey messages reliably and quickly. In his ground-breaking paper [2], Shannon introduced a model to describe communication. There is a *sender* trying to transmit a message to a *receiver* through an unreliable medium, known as *the channel*. In particular, he introduced the notion of *channel capacity* in [2]. Let the message be a sequence of k *bits*, and let the information rate be $r = \frac{k}{n}$ after adding $n - k$ redundant bits. By adding redundancy, Shannon proved that there is a fundamental trade-off between reliability and the information rate. When the information rate is below the channel capacity, the probability of error can be made arbitrarily small. This laid the foundation for *channel coding theory*.

Researchers have invested a great deal of time looking for good codes and practical decoding algorithms to achieve the channel capacity. The channel codes can be roughly divided into two classes: *algebraic* and *probabilistic*. For the algebraic method, powerful mathematical tools are employed to construct codes such that the code can have good algebraic structure, for example, large minimum distance $d_{\min}$. Bose-Chaudhuri-Hocquenghem (BCH) codes, Reed-Solomon (RS) codes and convolutional codes are some famous codes in this class. When the number of errors in the received bits is at most $\lfloor \frac{d_{\min}-1}{2} \rfloor$, elegant algebraic decoding algorithms guarantee the perfect recovery of messages. Instead of having an error-free recovery when the number of errors is at most $\lfloor \frac{d_{\min}-1}{2} \rfloor$, the probabilistic methods try to decode the message successfully with high probability. This relaxation allows the probabilistic method to correct many error patterns with more than $\lfloor \frac{d_{\min}-1}{2} \rfloor$ errors with high

probability.

The invention of turbo codes [3] by Berrou, Glavieux, and Thitimajshima is an important milestone in the probabilistic approach to decoding. The encoder of a turbo code encodes the same message with two simple codes. At the decoder, the received message is first decoded by one decoder, then the output is used as the input to the other decoder. The name "turbo" comes from the fact that the outputs of the decoders are passed between the decoders in a circular fashion, which is analogous to the mechanism of a "turbocharged" engine. By using the turbo algorithm, the performance of practical systems can approach Shannon's capacity limit. This result shows that one can get close to capacity with a moderate computational complexity. Later, it was observed that the turbo algorithm is an *iterative algorithm* similar to the one introduced by Gallager [4] in 1963. In that work, Gallager introduced another important class of codes, called *low-density parity-check* (LDPC) codes. However, these were not fully appreciated until being rediscovered by MacKay [5]. In Section 1.1, we briefly review some background on LDPC codes, which is needed for this dissertation. In this dissertation, we focus on the probabilistic method, and especially iterative decoding algorithms for LDPC and *generalized LDPC* (GLDPC) codes.

1.1 Background on LDPC Codes

In [4], Gallager proposed the class of regular LDPC codes. These are linear block codes defined by the null space of a sparse parity-check matrix. In the parity-check matrix, every column has the same number of non-zero entries, denoted by d_{v}, and every row has the same number of non-zero entries, denoted by d_{c}. A regular LDPC code can be obtained by randomly constructing a sparse matrix that satisfies these constraints. According to these constraints on the parity-check matrix, researchers

also denote the regular LDPC codes by $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes.

Besides defining a regular LDPC code, the sparse parity-check matrix also defines an undirected bipartite graph known as the Tanner graph of the code [6]. In this graph, there are two sets of nodes: the set of variable nodes and the set of check nodes. By treating the parity-check matrix as the adjacency matrix of the graph, rows in the parity-check matrix correspond to check nodes in the Tanner graph, and columns in the parity-check matrix correspond to variable nodes. For a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code, bit nodes and check nodes in the Tanner graph have degrees d_{v} and d_{c}, respectively. With the bipartite representation of the LDPC codes, decoding iteratively becomes a natural strategy for the design of decoders. Researchers have shown, empirically or theoretically, that the iterative algorithms work very well for the decoding of LDPC codes.

1.1.1 Irregular LDPC Codes

Various tools and techniques have also been proposed to improve the design of LDPC codes. One way to improve the design of LDPC codes is by relaxing the constraints of regular node degrees and allowing irregular *degree distributions* (d.d.). Instead of setting all variable node degrees to d_{v} and all check node degrees to d_{c}, Luby *et al.* [7] considered the case where both the variable nodes and the check nodes can have arbitrary degree profiles. With the same rate, it has been shown that the irregular LDPC codes can correct more errors than the regular LDPC codes [8]. Let λ_i be the fraction of edges in the graph that connecting to a variable node of degree i. Similar, ρ_i is the fraction of edges in the graph that connecting to a check node of degree i. Then, irregular LDPC codes can be represented by a pair of *edge-perspective* degree distributions, denoted by $(\lambda(x), \rho(x))$, where $\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i \geq 1} \rho_i x^{i-1}$. Equivalently, the d.d. of the irregular LDPC code can be

characterized from the *node perspective*. Let L_i and R_i be the fraction of variable nodes of degree i and the fraction of check nodes of degree i, respectively. The node-perspective d.d. pair, denoted by $(L(x), R(x))$, is defined by $L(x) = \sum_{i\geq 1} L_i x^i$ and $R(x) = \sum_{i\geq 1} R_i x^i$. Then, the design rate, denote by r, of LDPC codes can be computed using the d.d. pair by

$$r = 1 - \frac{L'(1)}{R'(1)} = 1 - \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}.$$

Density evolution (DE) is an important analysis technique for iterative algorithms, and was proposed by Richardson and Urbanke in [9]. For LDPC codes with asymptotically long codewords, the bit error probability of iterative decoding exhibits a *threshold* phenomena. The error probability has a sharp transition from 0 to 1 as the noise parameter increases. The location of the threshold for a LDPC code of a d.d. pair $(\lambda(x), \rho(x))$ can be computed by using DE analysis. It has been shown that, with a properly designed d.d. pair, the threshold of the irregular LDPC codes can be very close to the Shannon's limit of many channels. An extensive study and introduction of LDPC codes can be found in [10].

1.1.2 GLDPC Codes

Another way to improve the design of LDPC codes is replacing check nodes by a small component codes. The resulting code is called a GLDPC code. In the Tanner graph of a LDPC code, the check node represents a single-parity check (SPC) code. That is, variable nodes connected to the check node should form a bit vector with even parity. In 1981, Tanner generalized the SPC code at check nodes to some general small codes [6], such as small linear block codes. To emphasize the difference between LDPC codes and GLDPC codes, the check nodes in the graph are called *constraint nodes*. For a subset of variable nodes that are connected to a constraint

node in the Tanner graph, the valid bit vector for these variable nodes has to be a codeword of the small code at that constraint node.

After their introduction by Tanner, this class of codes remained largely unexplored until the work of Boutros *et al.*, [11] and Lentmaier and Zigangirov [12]. By introducing stronger constraint nodes than the SPC code to the graph, GLDPC codes can have both large minimum distance and good iterative decoding thresholds [12, 13]. However, when the soft-input soft-output (SISO) iterative decoding, such as belief-propagation (BP), is considered, the per-iteration decoding complexity for GLDPC codes is typically much higher than LDPC codes. This disadvantage of decoding complexity has slowed the adoption of GLDPC codes in practical designs.

Recently, GLDPC codes with degree-2 variable nodes have been widely considered for optical communication systems [14]. Codes in this class are called *generalized product* (GP) codes since product codes is a special case of codes in this class. From both the empirical and theoretical analysis, GP codes can provide high coding gains. But, if the component codes are optimally decoded, then the decoding complexity is still prohibitively high for very high-speed systems. Thus, finding low-complexity iterative decoding algorithms for GLDPC codes becomes an interesting topic in the channel coding area, and has been investigated by researchers [15, 13].

1.1.3 Spatially-Coupled Codes

The third method of improving the design of LDPC code is by *spatial coupling*. Convolutional LDPC codes were introduced by Felström and Zigangirov [16] in 1999, and have been observed to have an excellent performance. This surprising result spurred a large body of work, and has been shown that the threshold of the terminated convolutional LDPC codes approaches the capacity of many memoryless channels [17, 18]. Recently, Kudekar, Richardson, and Urbanke explained the mech-

anism behind this impressive performance in term of an operation, called *spatial coupling*. They also mathematically showed the existence of *threshold saturation* phenomena for the BEC channels. That is, as the coupled chain length increases, the BP threshold of the spatially-coupled LDPC codes will saturate to the maximum *a posteriori* (MAP) threshold of the underlying regular LDPC codes. Later, Yedla, the author, Nguyen and Pfister simplified the proof of the threshold saturation phenomena, and showed that the threshold saturation also exists for more general classes of scalar and vector recursions [19, 20]. With the same line of proof, Kumar *et al.*, also showed the threshold saturation phenomena for binary memoryless symmetric (BMS) channels [21].

Since the characterization of spatial coupling, applying spatial-coupling technique to the problems in various disciplines usually shows some performance improvement. It has been shown by Yedla [22] that spatially-coupled codes can achieve the universality of the multi-terminal communication system. Moreover, another notable example is to apply the spatial-coupling technique to compressed sensing problems. This problem was first investigated by Kudekar *et al.,* [23] and showed that the spatially-coupled sensing matrix can improve the sparsity to sampling threshold. Later, Donoho *et al.*, showed the threshold saturation phenomena also exists in compressed sensing problem when the approximate message passing (AMP) algorithm is employed to reconstruct signals [24]. In this dissertation, we apply spatial coupling to GP codes. The construction of a spatially-coupled GP code ensemble is introduced in Section 2.2.

1.2 Channel Coding and Inference Problems

The advent of iterative decoding algorithms started a revolution in several disciplines, including statistical physics, machine learning and signal processing [25, 26].

Some important problems, that are usually asked in these disciplines, are finding the marginal probabilities of the system, or finding the assignments which maximizes the marginal probability. In general, systems considered in these areas are very large. So, solving problems in these areas directly is usually infeasible. Fortunately, the system of interest can often be described by a sparse graph. Therefore, iterative algorithms, such as the BP algorithm and the max-product algorithm, are used to approximate the desired results. For graphs with cycles, it has been show that the correctness of the results obtained by these iterative algorithms are not guaranteed [27, 28]. But, the empirical results show that these iterative algorithms often work well and return good approximations.

Some of the algorithms studied in the inference problems are equivalent to the iterative algorithms in the area of channel coding. For example, the BP algorithm and the max-product algorithm is equivalent to the sum-product algorithm and the min-sum algorithm in the decoding algorithm of LDPC codes, respectively. Similar to the case of inference problem, the results computed by these iterative algorithms are not guaranteed to be correct, and checking the correctness of these solution is generally as hard as finding the correct solutions. Therefore, the second part of this dissertation study the iterative algorithm that the correctness of the result can be checked easily.

1.3 Dissertation Outline

This dissertation is organized as follows. In Section 2, we study a class of GLDPC codes, and observe that one can approach Shannon's limit in the high-rate regime by using iterative HDD. To the best of our knowledge, this is the first class of codes that can approach capacity using HDD. In Section 3, the weighted min-sum algorithm for regular LDPC codes are studied. The sufficient condition of the weights for the

convergence of the recursion is proposed. Also, conditions that prove the maximum likelihood (ML) optimality of the returned codeword are given. Finally, conclusions and future works are given in Section 4.

2. APPROACHING CAPACITY AT HIGH RATES WITH ITERATIVE HARD-DECISION DECODING

2.1 Introduction

In his groundbreaking 1948 paper, Shannon defined the capacity of a noisy channel as the largest information rate for which reliable communication is possible [2]. Since then, researchers have spent countless hours looking for ways to achieve this rate in practical systems. In the 1990s, the problem was essentially solved by the introduction of iterative soft decoding for turbo and low-density parity-check (LDPC) codes [3, 7, 29]. Although the decoding complexity is significant, these new codes were adopted quickly in wireless communication systems where the data rates were not too large [30, 31]. In contrast, complexity issues have slowed their adoption in very high-speed systems, such as those used in optical and wireline communication.

Introduced by Gallager in 1960, LDPC codes are linear block codes defined by a sparse parity-check matrix [32]. Using the parity-check matrix, an (N, K) LDPC code can be represented by a Tanner graph, which is a bipartite graph with N bit nodes and $N-K$ check nodes. The check nodes in the Tanner graph of an LDPC code represent the constraint that the group of bit nodes connected to a check node should form a codeword in a single-parity check (SPC) code. In 1981, Tanner generalized LDPC codes by replacing the SPC constraint nodes with more general constraints [6]. Particularly, the bit nodes connected to a check node are constrained to be codewords of (n, k) linear block codes such as Hamming codes, Bose-Chaudhuri-Hocquengham (BCH) codes or Reed-Solomon codes. After their introduction by

Tanner, generalized LDPC (GLDPC) codes remained largely unexplored until the work of Boutros *et al.* [11] and Lentmaier and Zigangirov [12].

GLDPC codes can have both large minimum distance and good iterative decoding thresholds [13]. But, the per-iteration decoding complexity of belief-propagation (BP) decoding of GLDPC codes is typically much higher than LDPC codes since optimal soft-input soft-output (SISO) decoding has to be performed for each component block code. However, the number of iterations required for the decoding algorithm to converge is substantially smaller. Recently, generalized product codes, *i.e.,* GLDPC codes with degree-2 bits, have been widely considered in optical communication systems [14]. In [33], GLDPC codes were proposed for 40Gb/s optical transport networks and it was shown that these codes outperform turbo codes by about 1 dB at a rate of 0.80. As such, GLDPC codes can provide high coding gains, but if the full BP decoder is used, the decoding complexity is still prohibitively high for implementation at very high-speed systems.

In this section, we show that by using iterative *hard-decision decoding* (HDD) of generalized product codes with BCH component codes, one can approach the capacity of the binary symmetric channel (BSC) in the high-rate regime. We consider an ensemble of spatially-coupled GLDPC codes based on t-error correcting BCH codes. For the BSC, we show that the redundancy-threshold tradeoff of this ensemble, under iterative HDD, scales optimally in the high-rate regime. To the best of our knowledge, this is the first example of an iterative HDD system that can provably approach capacity. It is interesting to note that iterative HDD of product codes was first proposed well before the recent revolution in iterative decoding but the performance gains were limited [34]. Iterative decoding of product codes became competitive only after the advent of iterative soft decoding based on the turbo principle [35, 36]. Then a slightly modified iterative HDD for GLDPC codes was proposed

by Miladinovic and Fossorier [13], and a good threshold performance of this iterative HDD algorithm for GLDPC codes were observed.

Under the assumption that the component code decoder corrects all patterns of t or fewer errors and leaves all other cases unchanged, the asymptotic noise threshold for product codes has been studied in [37, 38]. In [37], Schwartz *et al.* analyze the asymptotic block error probability for product codes using combinatorial arguments. By using random graph arguments, another asymptotic threshold analysis, based on the result of the existence of "k-core" in a random graph [39, 40], is proposed by Justesen *et al.* [38]. Finally, counting arguments are used in [41] to analyze the iterative HDD of GLDPC codes (without spatial coupling) for adversarial error patterns. Therefore, somewhat lower thresholds are reported.

Our choice of ensemble was motivated by the generalized product codes now used in optical communications [14] and their similarity to braided block codes [42, 43]. In particular, we consider the iterative HDD of generalized product codes with t-error correcting component codes. This is similar to other recent work on coding system for optical communication systems [44, 45, 46]. The main difference is that the proposed iterative HDD updates messages only using the extrinsic information. Therefore, HDD of our spatially-coupled GLDPC ensemble can be rigorously analyzed via density evolution (DE) even when miscorrection occurs. The DE analysis also allows us to show that iterative HDD can approach capacity in the high-rate regime. Also for generalized product codes, a practical implementation of the proposed iterative HDD is introduced.

2.2 Ensembles and Decoding Algorithms

In this section, various code ensembles and decoding algorithms are introduced. We first recall the GLDPC ensemble. Based on the GLDPC ensemble, the spatially-

coupled GLDPC ensemble is introduced. Also, a modified iterative HDD algorithm for GLDPC codes is proposed in this section. Since the proposed iterative HDD updates hard-decision messages only from extrinsic hard-decision messages, the performance of the proposed iterative HDD can be analyzed by DE. An ideal iterative HDD algorithm is also discussed, and its DE is described for the purpose of comparing with the proposed iterative HDD.

2.2.1 Ensembles

Let $\mathcal{C}$ be an $(n, k, d_{\min})$ binary linear code that can correct all error patterns of weight at most t (i.e., $d_{\min} \geq 2t + 1$). For example, one might choose $\mathcal{C}$ to be a primitive BCH code with parameters $(2^\nu - 1, 2^\nu - \nu t - 1, 2t + 1)$. Now, we consider a GLDPC ensemble where every bit node satisfies two code constraints defined by $\mathcal{C}$.

Definition 1. Each element of the $(\mathcal{C}, m)$ GLDPC ensemble is defined by a Tanner graph shown in Figure 2.1 and denoted by $\mathcal{G} = (\mathcal{I} \cup \mathcal{J}, \mathcal{E})$. There are $N = \frac{mn}{2}$ degree-2 bit nodes in set $\mathcal{I}$, and m degree-n code-constraint (or constraint) nodes defined by $\mathcal{C}$ in set $\mathcal{J}$. A random element from the ensemble is constructed by using an uniform random permutation for the mn edges from the bit nodes to the constraint nodes. From the construction of the code, one can show that the design rate of $(\mathcal{C}, m)$ ensemble is

$$R = \frac{N - m(n - k)}{N} = 1 - \frac{2(n - k)}{n} = 2\frac{k}{n} - 1.$$

Now, we consider a spatially-coupled GLDPC ensemble where every bit node satisfies two code constraints defined by $\mathcal{C}$. Similar to the definition introduced in [47], the spatially-coupled GLDPC ensemble $(\mathcal{C}, m, L, w)$ is defined as follows.

Definition 2. The Tanner graph of an element of the $(\mathcal{C}, m, L, w)$ spatially-coupled

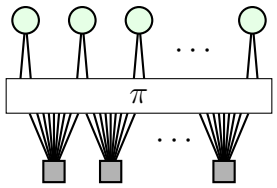

Figure 2.1: A $(\mathcal{C}, m)$ GLDPC ensemble, where π is a random permutation.

GLDPC contains L positions, $\{1, 2, \ldots, L\}$, of bit nodes and $L + w - 1$ positions, $\{1, 2, \ldots, L + w - 1\}$, of code-constraint nodes defined by $\mathcal{C}$. Let m be chosen such that mn is divisible by both 2 and w. At each position, there are $N = \frac{mn}{2}$ degree-2 bit nodes and m degree-n code-constraint nodes. A random element of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble is constructed as follows. At each bit position and code-constraint position, the mn sockets are partitioned into w groups of $\frac{mn}{w}$ sockets via a uniform random permutation. Let $\mathcal{S}_{i,j}^{(b)}$ and $\mathcal{S}_{i,j}^{(c)}$ be, respectively, the j-th group at the i-th bit position and the j-th group at i-th code-constraint position, where $j \in \{0, 1, \ldots, w - 1\}$. The Tanner graph is constructed by connecting $\mathcal{S}_{i,j}^{(b)}$ to $\mathcal{S}_{i+j,w-j-1}^{(c)}$ (*i.e.*, by mapping the $\frac{mn}{w}$ edges between the two groups). An example of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble with $w = 3$ is shown in Figure 2.2.

Remark 3. Since extra constraint nodes are required for spatial coupling, the design rate of the spatially-coupled ensemble is smaller than the design rate of the underlying ensemble [47]. According to the construction in Definition 2, $m(w-1)$ new constraint nodes are added after coupling. Thus, there are NL bit nodes and $m(L + w - 1)$

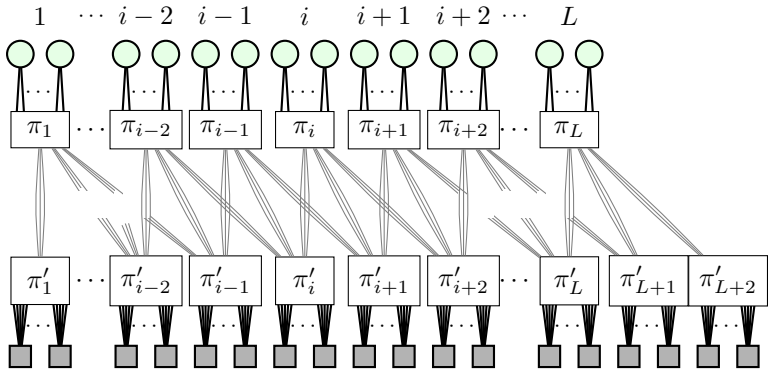

Figure 2.2: An example of $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble, where $w = 3$, and (π_i, π'_i) are random permutations at position i for bit nodes and constraint nodes, respectively.

constraint nodes in the spatially-coupled code. The resulting design rate is at least

$$\begin{aligned} R_{SC} &\geq \frac{NL - m(L+w-1)(n-k)}{NL} \\ &= 1 - \frac{2(n-k)}{n}\left(1 + \frac{w-1}{L}\right) \qquad (2.1) \\ &= R - (1-R)\frac{w-1}{L}, \qquad (2.2) \end{aligned}$$

where the second term in (2.2) is the rate loss due to adding constraint nodes. One can see that the rate loss goes to zero at a speed of $\frac{1}{L}$. We note that the actual rate, which is defined as the ratio of the dimension of the code and the codeword length, may be slightly higher due to the implied shortening of the code constraints.

2.2.2 Iterative HDD with Ideal Component Decoders

In this section, an iterative HDD with an ideal (i.e., genie aided) component-code decoder is introduced. This decoder corrects bits only when the number of error bits is less than or equal to the decoding radius t. In particular, the aid of a genie allows the ideal decoder to avoid miscorrection. To be explicit, we define

$\hat{\mathsf{D}} : \{0,1\}^n \times \{0,1\}^n \to \mathcal{C}$ as the operator of the ideal decoder. Given that a codeword $\boldsymbol{c} \in \mathcal{C}$ is transmitted, let $\boldsymbol{e} \in \{0,1\}^n$ be a binary error vector and $\boldsymbol{v} \triangleq \boldsymbol{c} \oplus \boldsymbol{e}$ be the received vector. Then, the output of the ideal decoder is

$$\hat{\mathsf{D}}(\boldsymbol{c}, \boldsymbol{e}) \triangleq \begin{cases} \boldsymbol{c} & \text{if } d_H(\boldsymbol{0}, \boldsymbol{e}) \leq t \\ \boldsymbol{c} \oplus \boldsymbol{e} & \text{otherwise,} \end{cases}$$

where $d_H(\cdot, \cdot)$ is the Hamming distance between the two arguments. Also, the bit-level mapping implied by the ideal decoder, denoted by $\hat{\mathsf{D}}_i : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, maps $(\boldsymbol{c}, \boldsymbol{e})$ to the i-th bit of $\hat{\mathsf{D}}(\boldsymbol{c}, \boldsymbol{e})$. The decoder performance is independent of the transmitted codeword since the component decoder satisfies the symmetry condition, *i.e.,* $\hat{\mathsf{D}}(\boldsymbol{c} \oplus \boldsymbol{c}', \boldsymbol{e}) = \hat{\mathsf{D}}(\boldsymbol{c}, \boldsymbol{e}) \oplus \boldsymbol{c}'$ for all $\boldsymbol{c}' \in \mathcal{C}$. In this case, the iterative HDD can be analyzed (e.g., using DE) under the assumption that the all-zero codeword is transmitted. Under this assumption, we can also define a simplified ideal decoding function $\hat{\mathsf{D}}(\boldsymbol{e}) \triangleq \hat{\mathsf{D}}(\boldsymbol{0}, \boldsymbol{e})$ that takes only one argument.

Now, we start to describe the ideal iterative HDD under the assumption that the all-zero codeword was transmitted. Decoding proceeds by passing binary messages along edges connecting variable nodes and constraint nodes. Let $r_i \in \{0,1\}$ denote the received channel value for the i-th variable node, and let $\nu_{i,j}^{(\ell)} \in \{0,1\}$ be the binary message from the i-th variable node to the j-th constraint node in the ℓ-th iteration. For simplicity, we assume no bit appears twice in a constraint, and let $\sigma_j(k)$ be the index of the variable node connected to the k-th socket of the j-th constraint. Let j' be the other neighbor of the i-th variable node, and $\sigma_j(k) = i$. Then, the iterative decoder is defined by the recursion

$$\nu_{i,j'}^{(\ell+1)} \triangleq \hat{\mathsf{D}}_k \left(\boldsymbol{v}_{i,j}^{(\ell)} \right),$$

where the candidate decoding vector for the j-th constraint node and i-th variable node is

$$\boldsymbol{v}_{i,j}^{(\ell)} \triangleq \left(\nu_{\sigma_j(1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(k-1),j}^{(\ell)}, r_i, \nu_{\sigma_j(k+1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(n),j}^{(\ell)}\right). \tag{2.3}$$

Note that the k-th entry is replaced by r_i. It is important to note that the above decoder passes extrinsic messages and *is not identical* to the conventional approach that simply iterates by exchanging the outputs of the component code decoders. In particular, replacing the k-th element by the received channel output enables rigorous DE analysis.

A *stopping set* is an error pattern where the messages associated with every component code either have 0 errors for greater than t errors. In this case, it is easy to verify that the conventional approach of running the ideal decoder for each component code results in no changes to the messages. With ideal component decoders, the final error pattern to which both decoders converge will be a stopping set. For the iterative HDD, we define the final error pattern as the set of bits where both component codes send a 1 (i.e., error) message. This odd convention follows from the fact that stopping sets arise somewhat more naturally in the context of erasure channels and decoding. On the other hand, the conventional approach with ideal component decoders and the above iterative HDD with ideal component decoders both return the same final error pattern after sufficiently many iterations. As we will see later, this equivalence does not hold when the component code decoders introduce miscorrections.

This decoding problem is also very closely connected to a well-known greedy algorithm for finding the k-core in a graph [39]. The k-core is the largest induced subgraph where all vertices have degree at least k. The connection can be seen by considering an *error graph* whose vertices are the code constraints where two vertices

are connected if there is a bit in both code constraints and that bit is received error. One can obtain this error graph from the Tanner graph by deleting all variable nodes associated with correctly received bits and then collapsing the remaining degree-2 variables nodes into edges that connect two constraint nodes. Therefore, the degree of a constraint vertex is equal to the number of errors in its attached bits and the error graph represents all errors and their associations with component codes. The greedy algorithm proceeds by removing any vertex of degree less than k along with all of its edges (because this cannot possibly be part of the k-core). Likewise, conventional iterative decoding with ideal component decoders can be seen as correcting the errors in a component code that contains fewer then $t+1$ errors. Therefore, stopping set found by iterative decoding is equivalent to the $(t+1)$-core of the error graph.

2.2.3 Density Evolution for Iterative HDD with Ideal Component Decoders

The iterative decoding performance of GLDPC codes can be analyzed via density evolution (DE) because, for a randomly chosen bit node, any fixed-depth neighborhood in the Tanner graph is a tree with high probability as $m \to \infty$. For HDD of the component codes, this DE can be written as a one-dimensional recursion.

For a randomly chosen edge (i, j') connecting a variable node i and a constraint node j', let $j \triangleq \mathcal{N}(i) \setminus j'$ be the other neighbor of the bit node i. One can see that the message passed on (i, j') is an error only when the channel output of the i-th bit is an error and there are at least t error inputs to the constraint node j. For a $n \in \mathbb{N}$, let $x^{(\ell)}$ be the error probability of a random chosen bit-to-constraint message in the ℓ-th iteration. The DE recursion of the iterative HDD with ideal component decoders for the $(\mathcal{C}, m)$ GLDPC ensemble is

$$x^{(\ell+1)} = p\hat{f}_n\left(x^{(\ell)}\right), \tag{2.4}$$

where

$$\hat{f}_n(x) \triangleq \sum_{i=t}^{n-1} \binom{n-1}{i} x^i \big(1-x\big)^{n-i-1}. \tag{2.5}$$

Let the noise threshold of the iterative HDD with ideal component decoders be defined by

$$\hat{p}_n^* \triangleq \sup\left\{ p \in (0,1] \,\middle|\, p\hat{f}_n(x) < x \text{ for } x \in (0,p] \right\}.$$

Similar to DE for LDPC codes on the BEC [10, pp. 95–96], there is a compact characterization of the hard-decision decoding threshold $\hat{p}_n^*$. Since $p\hat{f}_n(x)$ is monotone in p, the threshold $\hat{p}_n^*$ can be obtained by

$$\hat{p}_n^* = \inf_{x \in (0,1)} \frac{x}{\hat{f}_n(x)}. \tag{2.6}$$

Remark 4. This type of analysis is related to the threshold analysis for the k-core problem in [40] and [39]. Schwartz *et al.* also perform a combinatorial analysis in [37] to determine the decoding threshold for asymptotically long product codes. Their conclusion is somewhat different from the other analyses because they assume a finite number of decoding iterations and require that the block error rate vanishes. However, they treat the number of iterations explicitly and one can extract threshold estimates from [37, Cor. 2] in the limit as $r \to \infty$. In this case, a little algebra shows that the threshold c-value is $c^* = (t!)^{1/t}$. If the number of iterations is chosen to be $r > 2\log\log n$, then their equations imply that $p^* = c^*/n = (t!)^{1/t}/n$. Unfortunately, this last assumption violates some necessary conditions of their analysis and therefore gives only a lower bound on the correct threshold.

For the $(\mathcal{C}, m, L, w)$ ensemble, let $x_i^{(\ell)}$ be the average error probability of the hard-

decision messages emitted by bit nodes at position i in the ℓ-th iteration. Assume that $x_i^{(\ell)} = 0$ for all $i \notin \{1, 2, \ldots, L\}$ and $\ell \geq 0$. According to the construction in Definition 2, the average error probability of the hard-decision inputs to code-constraint nodes at position i is $y_i^{(\ell)} = \frac{1}{w} \sum_{j=0}^{w-1} x_{i-j}^{(\ell)}$. Then, the error probability of the hard-decision messages emitted by bit nodes at position i in the $(\ell+1)$-th iteration is

$$\begin{aligned} x_i^{(\ell+1)} &= \frac{1}{w} \sum_{k=0}^{w-1} p \hat{f}_n \left(y_{i+k}^{(\ell)} \right) \\ &= p \left(\frac{1}{w} \sum_{k=0}^{w-1} \hat{f}_n \left(\frac{1}{w} \sum_{j=0}^{w-1} x_{i-j+k}^{(\ell)} \right) \right). \end{aligned} \tag{2.7}$$

Remark 5. Note that (2.7) was first introduced by Kudekar *et al.* in [47], which tracks the average error probability of the output hard-decision messages from bit nodes at each position. One can obtain the DE update of the average error probability of the input hard-decision messages to the code-constraint nodes at the position $i \in \{1, 2, \ldots, L + w - 1\}$ by

$$\begin{aligned} y_i^{(\ell+1)} &= \frac{1}{w} \sum_{j=\max\{i-L,0\}}^{\min\{i-1,w-1\}} x_{i-j}^{(\ell+1)} \\ &= \frac{1}{w} \sum_{j=\max\{i-L,0\}}^{\min\{i-1,w-1\}} p \left(\frac{1}{w} \sum_{k=0}^{w-1} \hat{f}_n \left(y_{i-j+k}^{(\ell)} \right) \right). \end{aligned} \tag{2.8}$$

In the following analysis, we use (2.8) to find the noise threshold of the spatially-coupled system with iterative HDD. By the fact that, for any finite $w > 0$, $x_i^{(\ell)} \to 0$ for all $i \in \{1, 2, \ldots, L\}$ if and only if $y_i^{(\ell)} \to 0$ for all $i \in \{0, 1, \ldots, L + w - 1\}$, we know that the thresholds obtained from (2.7) and from (2.8) are identical.

2.2.4 Iterative HDD with Bounded Distance Decoders

It is well-known that GLDPC codes perform well under iterative soft decoding [35, 36]. The main drawback is that a posteriori probability (APP) decoding of the component codes can require significant computation. For this reason, we consider iterative HDD with bounded-distance decoding (BDD) of the component codes. Since the message update rule is the same as the rule introduced in Section 2.2.2, DE can also be employed to analyze the performance of this algorithm. In Section 2.6, a practical implementation of the iterative HDD algorithm is proposed.

Likewise, we start by defining the bit-level mapping implied by BDD, denoted by $\mathsf{D}_i : \{0,1\}^n \to \{0,1\}$, which maps the received vector $\boldsymbol{v} \in \{0,1\}^n$ to the i-th decoded bit according to the rule

$$\mathsf{D}_i(\boldsymbol{v}) = \begin{cases} c_i & \text{if } \boldsymbol{c} \in \mathcal{C} \text{ satisfies } d_H(\boldsymbol{c}, \boldsymbol{v}) \leq t \\ v_i & \text{if } d_H(\boldsymbol{c}, \boldsymbol{v}) > t \text{ for all } \boldsymbol{c} \in \mathcal{C}. \end{cases}$$

It is easy to verify that this decoder satisfies the symmetry condition, *i.e.,* $\mathsf{D}_i(\boldsymbol{v} \oplus \boldsymbol{c}) = \mathsf{D}_i(\boldsymbol{v}) \oplus c_i$ for all $\boldsymbol{c} \in \mathcal{C}$ and $i = 1, \ldots, n$.

We follow the same definition in Section 2.2.2. Let $r_i \in \{0,1\}$ denote the received channel value for variable node i and $\nu_{i,j}^{(\ell)} \in \{0,1\}$ be the binary message from the i-th variable node to the j-th constraint node in the ℓ-th iteration. The iterative decoder is defined by the recursion

$$\nu_{i,j'}^{(\ell+1)} \triangleq \mathsf{D}_k\left(\boldsymbol{v}_{i,j}^{(\ell)}\right), \tag{2.9}$$

where $\boldsymbol{v}_{i,j}^{(\ell)}$ is the candidate decoding vector for the j-th constraint node and the i-th variable node defined in (2.3). The corresponding DE analysis is introduced in the

following section.

2.2.5 Density Evolution for Iterative HDD with BDD

Since the component decoder is symmetric, then it suffices to consider the case where the all-zero codeword is transmitted over a BSC with error probability p [10, pp. 188–191]. Let $x^{(\ell)}$ be the error probability of the hard-decision messages passed from the variable nodes to the constraint nodes after ℓ iterations. For an arbitrary symmetric decoder, let $P_n(i)$ be the probability that a randomly chosen bit is decoded incorrectly when it is initially incorrect and there are i random errors in the other $n-1$ inputs. Likewise, let $Q_n(i)$ be the probability that a randomly chosen bit is decoded incorrectly when it is initially correct and there are i random errors in the other $n-1$ inputs. Then, for the $(\mathcal{C}, m)$ GLDPC ensemble, the DE recursion implied by (2.9) is defined by $x^{(0)} = p$,

$$x^{(\ell+1)} = f_n\left(x^{(\ell)}; p\right), \tag{2.10}$$

and (with $\overline{p} \triangleq 1-p$)

$$f_n(x;p) \triangleq \sum_{i=0}^{n-1} \binom{n-1}{i} x^i \big(1-x\big)^{n-i-1} \left(pP_n(i) + \overline{p}Q_n(i)\right). \tag{2.11}$$

For the iterative HDD with BDD described above, the quantities $P(i)$ and $Q(i)$ can be written in terms of the number of codewords of weight l in $\mathcal{C}$, denoted by A_l, [48]. Using the convention that $\binom{n}{k} = 0$ if $n < 0$, $k < 0$, or $k > n$, we define

$$l(i, \delta, j) \triangleq i - \delta + 2j + 1, \tag{2.12}$$

$$\Theta(n,i,\delta,j) \triangleq \binom{l(i,\delta,j)}{l(i,\delta,j)-j}\binom{n-l(i,\delta,j)-1}{\delta-1-j}\binom{n-1}{i}^{-1},$$

and

$$\Lambda(n,i,\delta,j) \triangleq \binom{l(i,\delta,j)-2}{l(i,\delta,j)-j-1}\binom{n-l(i,\delta,j)+1}{\delta-j}\binom{n-1}{i}^{-1}.$$

Since all decoding regions are disjoint, one can compute

$$P_n(i) = 1 - \sum_{\delta=1}^{t}\sum_{j=0}^{\delta-1}\frac{n-l(i,\delta,j)}{n}A_{l(i,\delta,j)}\Theta(n,i,\delta,j) \tag{2.13}$$

for $t \leq i \leq n-t-2$ and $P_n(i)=0$ for $0 \leq i \leq t-1$. Similarly,

$$Q_n(i) = \sum_{\delta=1}^{t}\sum_{j=0}^{\delta}\frac{l(i,\delta,j)-1}{n}A_{l(i,\delta,j)-1}\Lambda(n,i,\delta,j) \tag{2.14}$$

for $t+1 \leq i \leq n-t-1$, and $Q_n(i) = 0$ for $0 \leq i \leq t$. Note that, when the code contains the all-one codeword, $P_n(i) = 1$ for $n-t-1 \leq i \leq n-1$, and $Q_n(i) = 1$ for $n-t \leq i \leq n-1$.

Let the noise threshold be defined by

$$p_n^* \triangleq \sup\{p \in (0,1] \,|\, f_n(x;p) < x \text{ for all } x \in (0,p]\}. \tag{2.15}$$

According to the definition of p_n^*, one can show that $x^{(\ell)} \to 0$ as $\ell \to \infty$ for all $p < p_n^*$. By rewriting $f_n(x;p)$ as $f_n(x;p) = p[f_n(x;1) - f_n(x;0)] + f_n(x;0)$, we know that $f_n(x;p)$ is monotone in p. However, it is not clear to us whether or not $f_n(x;p)$ is monotone in x. Therefore, the definition of p_n^* does not imply that $\lim_{\ell\to\infty} x^{(\ell)} > 0$ for all $p > p_n^*$. Define $x^* \triangleq \sup\{z \in [0,1] \mid f_n(x;0) \leq x \text{ for all } x \in (0,z]\}$. We can

characterize p_n^* similar to (2.6) by

$$p_n^* = \inf_{x \in (0, x^*)} \frac{x - f_n(x; 0)}{f_n(x; 1) - f_n(x; 0)}.$$

Remark 6. Since the operations at bit nodes and constraint nodes are both sub-optimal, one may observe that for a fixed tuple (p, n, t), there exist some $x \in [0, 1]$ such that $f_n(x; p) > p$. This implies that the average error probability of the messages emitted by bit nodes after one iteration will be worse than the error probability of the channel output. In this case, bit nodes can just send the received channel bits to their neighbors. With this modification, the resulting DE update equation is

$$x^{(\ell+1)} = \min\{p, f_n(x^{(\ell)}; p)\}.$$

Let $q_n^* \triangleq \sup\{p \in (0, 1] \,|\min\{f_n(x; p), p\} < x \text{ for all } x \in (0, p]\}$ be the noise threshold of the modified decoding algorithm. We claim that $p_n^* = q_n^*$ by the following argument. From the fact that $\min\{p, f_n(x; p)\} \leq f_n(x; p)$, we have $q_n^* \geq p_n^*$. Consider the case of $p > p_n^*$. From (2.15), there exists some $x_0 \in (0, p]$ such that $f_n(x_0; p) \geq x_0$. Since $x_0 \in (0, p]$, one can show that $\min\{p, f_n(x_0; p)\} \geq x_0$ as well. Thus, we know $p > q_n^*$ from the definition of q_n^*. This implies that $q_n^* \leq p_n^*$, and therefore we conclude that $p_n^* = q_n^*$.

To derive the DE update equation of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble, let $x_i^{(\ell)}$ be the average error probability of hard-decision messages emitted by bit nodes at position i after the ℓ-th iteration. According to Definition 2, the average error probability of input messages to a code-constraint node at position i is $y_i^{(\ell)} = \frac{1}{w} \sum_{j=0}^{w-1} x_{i-j}^{(\ell)}$. It follows that $x_i^{(\ell+1)} = \frac{1}{w} \sum_{k=0}^{w-1} f_n(y_{i+k}^{(\ell)}; p)$ for $i \in \{1, 2, \ldots, L\}$, where $f_n(x; p)$ is defined in (2.11). We also set $x_i^{(\ell)} = 0$ for $i \notin \{1, 2, \ldots, L\}$. There-

fore, in the $(\ell+1)$-th iteration, the average error probability of the hard-decision messages emitted by bit nodes at the position $i \in \{1, 2, \ldots, L\}$ is given by

$$x_i^{(\ell+1)} = \frac{1}{w} \sum_{k=0}^{w-1} f_n \left(\frac{1}{w} \sum_{j=0}^{w-1} x_{i-j+k}^{(\ell)}; p \right). \tag{2.16}$$

Similar to the discussion in Remark 5, the DE update of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble which tracks the average error probability of the input messages to the constraint nodes at the position $i \in \{1, 2, \ldots, L+w-1\}$ can also be obtained by

$$\begin{aligned} y_i^{(\ell+1)} &= \frac{1}{w} \sum_{j=\max\{i-L,0\}}^{\min\{i-1,w-1\}} x_{i-j}^{(\ell+1)} \\ &= \frac{1}{w} \sum_{j=\max\{i-L,0\}}^{\min\{i-1,w-1\}} \frac{1}{w} \sum_{k=0}^{w-1} f_n \left(y_{i-j+k}^{(\ell)}; p \right). \end{aligned}$$

2.3 BCH Component Codes

In the remainder of this section, an $(n, k, 2t+1)$ binary primitive BCH code (or its $(n, k-1, 2t+2)$ even-weight subcode) will be used as the component code for both the $(\mathcal{C}, m)$ GLDPC and $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensembles. When the exact weight spectrum is known, one can compute $P_n(i)$ and $Q_n(i)$ using (2.13) and (2.14), respectively. Otherwise, we use the asymptotically-tight binomial approximation

$$A_l = \begin{cases} 2^{-\nu t} \binom{n}{l} \left(1 + O\left(n^{-0.1}\right)\right) & \text{if } d \le l \le n-d, \\ 1, & \text{if } l = 0,\ l = n, \\ 0, & \text{otherwise,} \end{cases} \tag{2.17}$$

for $n \geq n_t$, where $d = 2t + 1$, $n = 2^\nu - 1$ and n_t is a constant depends on t [49].

For the $(n, k-1, 2t+2)$ even-weight subcode of an $(n, k, 2t+1)$ primitive BCH code, the number of codewords is denoted by $\tilde{A}_l$ where $\tilde{A}_l = A_l$ when l is even and $\tilde{A}_l = 0$ when l is odd. Let $\tilde{P}_n(i)$ and $\tilde{Q}_n(i)$ be the miscorrection probabilities implied by $\tilde{A}_l$ for the even-weight subcode. Similar to $P_n(i)$ and $Q_n(i)$ in the $(n, k, 2t+1)$ primitive BCH code, it can be shown that $\tilde{P}_n(i) = 0$ for $0 \leq i \leq t-1$ and $\tilde{Q}_n(i) = 0$ for $0 \leq i \leq t+1$. Then, the DE recursions for the $(\mathcal{C}, m)$ GLDPC ensemble and the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble can be obtained from (2.10) and (2.16), respectively.

2.3.1 High-Rate Scaling Limit for Iterative HDD with Ideal Component Decoders

In [38, 45], Justesen *et al.* analyze the asymptotic performance of long product codes under the assumption that the component decoders have no miscorrection. These arguments can be applied for the decoding of both BSC and BEC outputs. By considering the decoding process as removing vertices of degree less or equal to t, they show that the process fails if the error graph contains $(t+1)$-core. The problem of having "k-cores" in a random graph has attracted considerable interests in graph theory [39, 40]. By employing the results in [39], Justesen *et al.* characterize the evolution for the number of errors per constraint node as a recursion for the "Poisson parameter". That recursion leads to a threshold for successful decoding on the average number of error bits attached to a code-constraint node. In this section, we derive the high-rate scaling limiting recursion of the proposed iterative algorithm with ideal HDD. Then, we show that the obtained high-rate scaling limit (2.18) has the same update equation as the recursion of Poisson parameter in [38, 45].

For a fixed $\rho > 0$, let $p \triangleq \frac{\rho}{n-1}$ scale with n and $\lambda_n \triangleq (n-1)x^{(\ell)}$. The recursion

(2.4) for $\lambda_n^{(\ell)}$ becomes

$$\lambda_n^{(\ell+1)} = (n-1)p\hat{f}_n\left(\frac{\lambda_n^{(\ell)}}{n-1}\right) = \rho\hat{f}_n\left(\frac{\lambda_n^{(\ell)}}{n-1}\right).$$

For any $\ell > 0$, let $\lambda^{(\ell)} \triangleq \lim_{n\to\infty} \lambda_n^{(\ell)}$. The high-rate scaling limit of the recursion for the ideal component code decoder is

$$\lambda^{(\ell+1)} = \lim_{n\to\infty} \rho\hat{f}_n\left(\frac{\lambda_n^{(\ell)}}{n-1}\right) \triangleq \hat{f}\left(\lambda^{(\ell)};\rho\right).$$

Define the tail probability of the Poisson distribution with mean λ by

$$\phi(\lambda;k) \triangleq \sum_{i=k+1}^{\infty} \frac{\lambda^i}{i!}e^{-\lambda}.$$

By the Poisson theorem [50, pp. 113], one can show that $\lim_{n\to\infty} \rho\hat{f}_n(\frac{\lambda}{n-1}) = \rho\phi\left(\lambda;t-1\right)$. Thus, the high-rate scaling limit of the recursion for the ideal component code decoder becomes

$$\lambda^{(\ell+1)} = \rho\phi\left(\lambda^{(\ell)};t-1\right). \tag{2.18}$$

The scaled noise threshold $\hat{\rho}^*$ is the largest ρ such that the iteration converges from ρ to 0 and is defined by

$$\hat{\rho}^* \triangleq \sup\left\{\rho \in [0,\infty) \mid \rho\phi\left(\lambda;t-1\right) < \lambda \text{ for } \lambda \in (0,\rho]\right\}.$$

Since $\phi(\lambda;t-1)$ is increasing and upper bounded by 1, it follows that

$$\hat{\rho}^* = \inf_{\lambda>0} \frac{\lambda}{\phi\left(\lambda;t-1\right)}$$

Remark 7. This threshold condition is identical to the ones given in [39] and [40] for the equivalent k-core problem. The connection to asymptotically long product codes with ideal bounded distance decoding was first made in [38].

For the spatially-coupled GLDPC ensemble, let $\lambda_i^{(\ell)}$ with $i \in \{1, 2, \ldots, L\}$ be the average number of error messages emitted by bit nodes at position i in the ℓ-th iteration. We set $\lambda_i^{(0)} = \rho$ for all $i \in \{1, 2, \ldots, L\}$, and set $\lambda_i^{(\ell)} = 0$ for all $i \notin \{1, 2, \ldots, L\}$ and $\ell \geq 0$. The recursion for the spatially-coupled ensemble is

$$\begin{aligned} \lambda_i^{(\ell+1)} &= \frac{1}{w} \sum_{k=0}^{w-1} \hat{f} \left(\frac{1}{w} \sum_{j=0}^{w-1} \lambda_{i-j+k}^{(\ell)}; \rho \right) \\ &= \rho \left(\frac{1}{w} \sum_{k=0}^{w-1} \phi \left(\frac{1}{w} \sum_{j=0}^{w-1} \lambda_{i-j+k}^{(\ell)}; t-1 \right) \right). \end{aligned} \tag{2.19}$$

Remark 8. We note that this vector update equation is the same as the vector update equation used for Q-coloring analysis (with $Q = t + 1$) of spatially-coupled graphs in [51]. Therefore, the results in Section 2.4.1 also apply to the Q-coloring problem.

2.3.2 High-Rate Scaling Limit for Iterative HDD with BDD

We have shown that the recursion using the random graph argument in [38] and [45] is the same as the DE analysis of ideal iterative HDD in the limit as $n \to \infty$. The main weakness of the random graph argument is that it is not applicable to practical decoders. In this section, the high-rate scaling limit of the recursion for our DE analysis as $n \to \infty$ is introduced. The main contribution is that our approach rigorously accounts for miscorrection.

We first introduce some notation and a few lemmas that simplify the development. Consider the Poisson distribution with mean λ. Let $\psi(\lambda; k)$ and $\varphi(\lambda; k)$ be, respectively, the tail probability for the even terms, and the tail probability for the

odd terms. Then, we have

$$\psi(\lambda;k) \triangleq \frac{1+e^{-2\lambda}}{2} - \sum_{i=0}^{\lfloor k/2 \rfloor} \frac{\lambda^{2i}}{(2i)!} e^{-\lambda},$$
$$\varphi(\lambda;k) \triangleq \frac{1-e^{-2\lambda}}{2} - \sum_{i=0}^{\lfloor k/2 \rfloor} \frac{\lambda^{(2i+1)}}{(2i+1)!} e^{-\lambda}.$$

Lemma 9. *For the codes described above and $t \leq i \leq n-t-2$, the limit $\lim_{n\to\infty} P_n(i) = 1$. Also, for the same code and $t+1 \leq i \leq n-t-1$, the function $nQ_n(i)$ is bounded. If $\lfloor \sqrt{n} \rfloor > t+1$, then*

$$nQ_n(i) \leq \frac{1}{(t-1)!} + O\left(n^{-0.1}\right) \tag{2.20}$$

for all $t+1 \leq i \leq \lfloor \sqrt{n} \rfloor$. Thus, for any fixed $i > 0$, the limit $\lim_{n\to\infty} nQ_n(i) = \frac{1}{(t-1)!}$.

Proof. See Appendix A.1

Consider the DE recursion (2.10) for the $(\mathcal{C}, m)$ GLDPC ensemble. For a fixed ρ, let $p \triangleq \frac{\rho}{n-1}$ scale with n and $\lambda_n^{(\ell)} \triangleq (n-1)x^{(\ell)}$. From (2.10) and (2.11), the recursion for $\lambda_n^{(\ell)}$ equals

$$\begin{aligned}\lambda_n^{(\ell+1)} &= (n-1) f_n\left(\frac{\lambda_n^{(\ell)}}{n-1}; \frac{\rho}{n-1}\right) \\ &= \sum_{i=t}^{n-1} \binom{n-1}{i} \left(\frac{\lambda_n^{(\ell)}}{n-1}\right)^i \left(1 - \frac{\lambda_n^{(\ell)}}{n-1}\right)^{n-1-i} \\ &\quad \times \left(\rho\left(P_n(i) - Q_n(i)\right) + (n-1)\, Q_n(i)\right), \end{aligned} \tag{2.21}$$

with initial value $\lambda_n^{(0)} = \rho$ for all n.

Lemma 10. *Let $X_n \sim B(n-1, \frac{\lambda_n}{n-1})$ be a sequence of binomial random variables for $n-1$ trials with success probability $\frac{\lambda_n}{n-1}$. If $\lambda_n \to \lambda < \infty$, then $\lim_{n\to\infty} E\left[P_n(X_n)\right] =$*

$\phi(\lambda; t-1)$ *and* $\lim_{n\to\infty} E[Q_n(X_n)] = 0$.

Proof. See Appendix A.2.

Lemma 11. *Let* $X_n \sim B(n-1, \frac{\lambda_n}{n-1})$ *be a sequence of binomial random variables for* $n-1$ *trials with success probability* $\frac{\lambda_n}{n-1}$*. If* $\lambda_n \to \lambda < \infty$*, then*

$$\lim_{n\to\infty} E[nQ_n(X_n)] = \frac{\phi(\lambda; t)}{(t-1)!}.$$

Proof. See Appendix A.3.

Using Lemma 10 and Lemma 11, one can simplify the recursion for $\lambda^{(\ell)} \triangleq \lim_{n\to\infty} \lambda_n^{(\ell)}$.

Lemma 12. *For any fixed* $\ell > 0$*, the limit* $\lambda^{(\ell)} \triangleq \lim_{n\to\infty} \lambda_n^{(\ell)}$ *exists, and the recursion for* $\lambda^{(\ell)}$ *is given by* $\lambda^{(0)} = \rho$ *and*

$$\begin{aligned}\lambda^{(\ell+1)} &= f\left(\lambda^{(\ell)}; \rho\right) \\ &\triangleq \rho\phi\left(\lambda^{(\ell)}; t-1\right) + \frac{1}{(t-1)!}\phi\left(\lambda^{(\ell)}; t\right). \end{aligned} \tag{2.22}$$

Proof. We prove the lemma by induction. The base case $\lambda^{(0)} = \rho$ holds by assumption. For the inductive step, suppose that $\lambda^{(\ell)} = \lim_{n\to\infty} \lambda_n^{(\ell)}$ exists. Let $X_n^{(\ell)} \sim \mathrm{B}(n-1, \frac{\lambda_n^{(\ell)}}{n-1})$ be a binomial random variable with parameters $n-1$ and $\frac{\lambda_n^{(\ell)}}{n-1}$. Then, the recursion (2.21) can be represented as $\lambda_n^{(\ell+1)} = E[\rho(P_n(X_n^{(\ell)}) - Q_n(X_n^{(\ell)})) + (n-1)Q_n(X_n^{(\ell)})]$. Again, Lemma 10 and Lemma 11 imply that $\lim_{n\to\infty} E[P_n(X_n^{(\ell)})]$, $\lim_{n\to\infty} E[Q_n(X_n^{(\ell)})]$, and $\lim_{n\to\infty} E[nQ_n(X_n^{(\ell)})]$ exist. Thus, the limit of $\lambda_n^{(\ell+1)}$ exists as $n \to \infty$, and satisfies the recursion

$$\lambda^{(\ell+1)} \triangleq \lim_{n\to\infty} \lambda_n^{(\ell+1)} = \rho\phi\left(\lambda^{(\ell)}; t-1\right) + \frac{1}{(t-1)!}\phi\left(\lambda^{(\ell)}; t\right).$$

This completes the mathematical induction.

Remark 13. For any $n < \infty$, $\frac{n}{n-1}\rho$ can be seen as the average number of initial error bits attached to a code-constraint node, and $\frac{n}{n-1}\lambda_n^{(\ell)}$ can be viewed as the average number of error messages passed to a code-constraint node after the ℓ-th iteration. Since $\frac{n}{n-1}\lambda_n^{(\ell)} \to \lambda^{(\ell)}$, it follows that the recursion (2.22) tracks the evolution of the average number of error messages passed to a code-constraint node.

The following lemma shows that the DE recursion for the GLDPC ensemble whose component code is the even-weight subcode of a BCH code can be obtained by modifying (2.22).

Lemma 14. *Consider the GLDPC ensemble whose component code is the even-weight subcode of a BCH code. If t is even, then the recursion for $\lambda^{(\ell)}$ is $\lambda^{(\ell+1)} = \tilde{f}_e\left(\lambda^{(\ell)}; \rho\right)$ with*

$$\tilde{f}_e\left(\lambda^{(\ell)}; \rho\right) \triangleq \rho\phi\left(\lambda^{(\ell)}; t-1\right) + \frac{1}{(t-1)!}\psi\left(\lambda^{(\ell)}; t\right).$$

If t is odd, the recursion is $\lambda^{(\ell+1)} = \tilde{f}_o\left(\lambda^{(\ell)}; \rho\right)$ with

$$\tilde{f}_o\left(\lambda^{(\ell)}; \rho\right) \triangleq \rho\phi\left(\lambda^{(\ell)}; t-1\right) + \frac{1}{(t-1)!}\varphi\left(\lambda^{(\ell)}; t\right).$$

Proof. See Appendix A.4.

For the spatially-coupled GLDPC ensemble, let $\lambda_i^{(\ell)}$ with $i \in \{1, 2, \ldots, L\}$ be the average number of error messages emitted by bit nodes at positions i in the ℓ-th iteration. We set $\lambda_i^{(0)} = \rho$ for all $i \in \{1, 2, \ldots, L\}$, and set $\lambda_i^{(\ell)} = 0$ for all $i \notin \{1, 2, \ldots, L\}$ and $\ell \geq 0$. Similar to (2.16), the recursion for spatially-coupled

ensemble is

$$\lambda_i^{(\ell+1)} = \frac{1}{w}\sum_{k=0}^{w-1} f\left(\frac{1}{w}\sum_{j=0}^{w-1}\lambda_{i-j+k}^{(\ell)};\rho\right) \tag{2.23}$$

for $i \in \{1, 2, \ldots, L\}$. When the even-weight subcode of a BCH code is used as a component code in the spatially-coupled GLDPC ensemble, the recursion becomes

$$\lambda_i^{(\ell+1)} = \begin{cases} \dfrac{1}{w}\displaystyle\sum_{k=0}^{w-1} \tilde{f}_e\left(\frac{1}{w}\sum_{j=0}^{w-1}\lambda_{i-j+k}^{(\ell)};\rho\right) & \text{if } \ t \text{ is even,} \\ \dfrac{1}{w}\displaystyle\sum_{k=0}^{w-1} \tilde{f}_o\left(\frac{1}{w}\sum_{j=0}^{w-1}\lambda_{i-j+k}^{(\ell)};\rho\right) & \text{if } \ t \text{ is odd.} \end{cases}$$

2.4 Bounds on the Noise Threshold

In this section, we consider the noise thresholds of the decoding algorithms in Section 2.2 for the spatially-coupled ensemble when $L \gg w$ and $w \to \infty$. We employ the analysis proposed by Yedla *et al.* [19, 20] to compute these thresholds. Consider the recursion defined by the DE update equation of a decoding algorithm. Let x be the average error probability of the messages emitted by bit nodes. For each channel parameter $p \in [0, 1]$, the *potential function*, denoted by $U(x; p)$, is proposed in [19], and the *energy gap* $\Delta E(p)$ is defined as the minimum of the potential function over the non-zero fixed points of the recursion. If the recursion is a *scalar admissible system* (see [19, Definition 1]), the *potential threshold*, p^{**}, defined as the supremum of the channel parameter p such that $\Delta E(p) \geq 0$ is introduced. Finally, they show that the noise threshold of the decoding algorithm saturates to p^{**} when the algorithm is applied to a spatially-coupled system with $L \gg w$ and $w \to \infty$ [19, 20].

While the exact value of p^{**} is easily computed numerically, a simple expression is not readily available. Therefore, we derive instead a suitable lower bound.

2.4.1 Iterative HDD with Ideal Component Decoders

Suppose that one ignores the effect of miscorrection and considers the natural hard-decision peeling decoder for the $(\mathcal{C}, m)$ ensemble based on BCH codes, then it is easy to see that at most mt errors can be corrected using BDD. To achieve this upper bound, it must happen that each code corrects exactly t errors. If some codes decode with fewer than t errors, then there is an irreversible loss of error-correcting potential. Since there are $\frac{nm}{2}$ code bits per code constraint, normalizing this number shows that the noise threshold is upper bounded by $\frac{2t}{n}$. In terms of the average number of errors in each code constraint, the threshold is upper bounded by $2t$ because each code involves n bits.

Before we delve into the analysis of the potential threshold for the iterative HDD with ideal component decoders, we first recall that the definitions of the beta function, $B(a, b)$, and the normalized incomplete beta function, $I_x(a, b)$, [52, §8.17] are given by

$$B(a, b) \triangleq \int_0^1 z^{a-1}(1-z)^{b-1}dz \tag{2.24}$$

and

$$I_x(a, b) \triangleq \frac{1}{B(a, b)} \int_0^x z^{a-1}(1-z)^{b-1}dz \triangleq \frac{B_x(a, b)}{B(a, b)}. \tag{2.25}$$

When a, b are positive integers, these functions have the following well-known prop-

erties [52, §8.17]:

$$B(a,b) = \frac{(a-1)!(b-1)!}{(a+b-1)!},$$

$$I_x(a,b) = \sum_{k=a}^{a+b-1} \binom{a+b-1}{k} x^k (1-x)^{a+b-1-k}, \tag{2.26}$$

$$B(a+1,b) = \frac{a}{a+b} B(a,b), \tag{2.27}$$

$$I_x(a+1,b) = I_x(a,b) - \frac{x^a(1-x)^b}{aB(a,b)}. \tag{2.28}$$

Since the DE update equation (2.4) for the iterative HDD without miscorrection is a scalar recursion. The following lemma enables us to apply the result in [19] to the current noise threshold analysis.

Lemma 15. *Let the recursion (2.4) be represented as* $x^{(\ell+1)} = \tilde{f}(\tilde{g}(x^{(\ell)}); p)$ *with* $\tilde{f}(x;p) = px$ *and* $\tilde{g}(x) = \hat{f}_n(x)$*. The pair* $(\tilde{f}, \tilde{g}) = (px, \hat{f}_n(x))$ *is a scalar admissible system for any finite* $n > 0$*.*

Proof. It is obvious that $\tilde{f}(x;p)$ is strictly increasing in both x and p, and $\tilde{f}(0;p) = \tilde{f}(x;0) = 0$. Using (2.25), we rewrite $\tilde{g}(x)$ by

$$\tilde{g}(x) = \hat{f}_n(x) = \frac{1}{B(t, n-t)} \int_0^x z^{a-1}(1-z)^{b-1} dz,$$

then we know $\tilde{g}(0) = 0$. Since $\tilde{g}'(x) \triangleq \frac{\mathrm{d}\tilde{g}(x)}{\mathrm{d}x} = \frac{1}{B(t,n-t)} x^{a-1}(1-x)^{b-1} > 0$ for all $x \in (0,1)$, $\tilde{g}(x)$ is strictly increasing in x. Also, the second derivative

$$\frac{\mathrm{d}^2\tilde{g}(x)}{\mathrm{d}x^2} = \frac{1}{B(t,n-t)} \left((a-1)x^{a-2}(1-x)^{b-1} - (b-1)x^{a-1}(1-x)^{b-2}\right)$$

is a continuous function. Therefore, the recursion (2.4) is a scalar admissible system according to [19, Definition 1].

By the fact that the DE update (2.4) defines a scalar admissible system for any $n > 0$, the associated potential function is given by the following definition.

Definition 16. The *potential function* for the recursions (2.4) is defined by $V_n(x;p) \triangleq \int_0^x (z - p\hat{f}_n(z))\hat{f}_n'(z)\mathrm{d}z$. The *potential threshold* for the recursion (2.4) is

$$\hat{p}_n^{**} \triangleq \sup\left\{p \in [0,1] \middle| \min_{x\in(0,1]} V_n(x;p) \geq 0\right\}. \tag{2.29}$$

This threshold for iterative HDD without miscorrection is achieved by $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensembles in the limit where $m \gg L \gg w$ as $w \to \infty$ [19, 20].

Remark 17. Let $K_{\tilde{f},\tilde{g}}$ be a constant determined by the system $(\tilde{f}, \tilde{g})$ [19, Lemma 5]. In [19, Theorem 1], Yedla *et al.* showed that, for all $p < \hat{p}_n^{**}$ and $w > K_{\tilde{f},\tilde{g}}/\Delta E(p)$, the error probabilities $y_i^{(\ell)}$ for all $i \in \{1, 2, \ldots, L + w - 1\}$ in (2.8) converge to 0 as $\ell \to \infty$. Moreover, this implies that $\hat{p}_n^{**}$ is also a noise threshold for the recursion (2.7) according to the fact that $x_i^{(\ell)} \to 0$ for all $i \in \{1, 2, \ldots, L\}$ if and only if $y_i^{(\ell)} \to \infty$ for all $i \in \{1, 2, \ldots, L + w - 1\}$.

Remark 18. Since $V_n(x;p) = x\hat{f}_n(x) - \int_0^x \hat{f}_n(z)\mathrm{d}z - \frac{p}{2}\hat{f}_n^2(x)$, we know that $V_n(x;p)$ is decreasing in p. Therefore, one can easily obtain $\hat{p}_n^{**}$ by numerically computing

$$\hat{p}_n^{**} = \inf_{x\in(0,1]} \frac{x\hat{f}_n(x) - \int_0^x \hat{f}_n(z)\mathrm{d}z}{\frac{1}{2}\hat{f}_n^2(x)}.$$

As shown in [19], for any $p \in [0, 1]$, fixed points of the recursion (2.4) are corresponding to stationary points of $V_n(x;p)$. Since the minimum of the potential function always occurs at a stationary point which is corresponding to a stable fixed point [20], it is sufficient to consider the potentials at fixed points. For a $p \in (0, 1]$, let x be a non-zero fixed point of the recursion (2.4) with $n > 0$. Then, we have

$x = p\hat{f}_n(x)$, and define $p(x) \triangleq \frac{x}{\hat{f}_n(x)}$. Since $\hat{f}_n(x)$ is increasing in x, the pair $(x, p(x))$ for all $x \in (0, 1]$ characterizes all possible non-zero fixed points for the recursion (2.4) for all $x \in (0, 1]$ and $p \in (0, 1]$. Thus, we can have the fixed-point potentials as a function of $x \in (0, 1]$. The potential at the fixed point x, denoted by $V_n(x)$, is

$$V_n(x) \triangleq V_n\left(x; p\left(x\right)\right) = x\hat{f}_n\left(x\right) - 2\int_0^x \hat{f}_n(z)\mathrm{d}z.$$

Remark 19. $V_n(x)$ is the potential at the fixed point of the admissible system $(\tilde{f}, \tilde{g}) = (px, \hat{f}_n(x))$, and the DE update equation for the corresponding spatially-coupled system is given in (2.8). On the other hand, one can also consider the update equation (2.4) as the recursion of the function pair $(\tilde{f}, \tilde{g}) = (p\hat{f}_n(x), x)$. Then, the DE update equation for the corresponding spatially-coupled system is

$$y_i^{(\ell+1)} = \frac{1}{w}\sum_{j=\max\{i-L,0\}}^{\min\{i-1,w-1\}} p\hat{f}_n\left(\frac{1}{w}\sum_{k=0}^{w-1} y_{i-j+k}^{(\ell)}\right),$$

where $y_i^{(\ell)}$ is the average error probability of the input messages to the $\tilde{g}$-nodes at the i-th position and $i \in \{1, 2, \ldots, L + w - 1\}$. Using the proof of Lemma 15, one can show that $(p\hat{f}_n(x), x)$ is also a scalar admissible system. By the definition of the potential function in [19, Definiton 2], the potential function of the $(p\hat{f}_n(x), x)$ system is $\hat{U}_n(x; p) \triangleq \int_0^x (z - p\hat{f}_n(z))\mathrm{d}z$. This is the potential function introduced in [53]. Let $\hat{U}_n(x) \triangleq \hat{U}_n(x; p(x))$ be the potential of the $(p\hat{f}_n(x), x)$ system at the fixed point $(x, p(x))$. Since $(px, \hat{f}_n(x))$ and $(p\hat{f}_n(x), x)$ systems share the same set of fixed points, one can show that their potentials at non-zero fixed points satisfy

$$V_n(x) = \frac{2\hat{f}_n(x)}{x}\hat{U}_n(x).$$

To find the lower bound of the potential threshold, it is convenient to represent the potential functions by the incomplete beta function. From (2.5) and (2.26), $V_n(x)$ can be written as

$$V_n(x) = xI_x(t, n-t) - 2\int_0^x I_z(t, n-t)\mathrm{d}z.$$

Applying integration by parts, we have

$$\begin{aligned} V_n(x) &= xI_x(t, n-t) - 2zI_z(t, n-t)\Big|_0^x + 2\int_0^x z\mathrm{d}I_z(t, n-t) \\ &= -xI_x(t, n-t) + \frac{2}{B(t, n-t)}\int_0^x zz^{t-1}(1-z)^{n-t-1}dz. \end{aligned}$$

By (2.25), one can rewrite $V_n(x)$ as

$$V_n(x) = -xI_x(t, n-t) + \frac{2B_x(t+1, n-t)}{B(t, n-t)}.$$

By (2.27) and (2.28), one gets

$$\begin{aligned} V_n(x) &= -xI_x(t, n-t) + \frac{2t}{n}\frac{B_x(t+1, n-t)}{B(t+1, n-t)} \\ &= -xI_x(t, n-t) + \frac{2t}{n}\left(I_x(t, n-t) - \frac{x^t(1-x)^{n-t}}{tB(t, n-t)}\right) \\ &= -\left(x - \frac{2t}{n}\right)I_x(t, n-t) - \frac{2x^t(1-x)^{n-t}}{nB(t, n-t)}. \end{aligned} \tag{2.30}$$

Using (2.30), we can lower bound the potential threshold $\hat{p}^{**}$ with the following lemma.

Lemma 20. *Let $\hat{x}_n^{**} \in (0, 1]$, and let $(\hat{x}_n^{**}, \hat{p}_n^{**})$ be a fixed point of (2.4) and $V_n(\hat{x}_n^{**}) = 0$. Then, there exists a pair (n_0, t_0) such that, for all $n \geq n_0$ and $t \geq t_0$, $\frac{2t-2}{n} \leq \hat{x}_n^{**} \leq \frac{2t}{n}$. Moreover, $\hat{p}_n^{**} > \frac{2t-2}{n}$.*

Proof. See Appendix A.5.

Remark 21. The constants t_0 and n_0 are for a rigorous proof of $V_n(\frac{2t-2}{n}) > 0$ when $n \geq n_0$ and $t \geq t_0$. However, through the numerical evaluation, we observed that $V_n(\frac{2t-2}{n}) > 0$ for all $t \geq 2$ and $n > 2t$. Thus, we conjecture that the lemma holds for all $t \geq 2$ and $n > 2t$.

For the high-rate scaling limit recursion (2.18), one can show that $\hat{f}(\lambda; \rho) = \rho\phi(\lambda; t-1)$ is increasing in both λ and ρ. Thus, the noise threshold in terms of the average number of errors per $n-1$ bits for the recursion (2.18), denoted by $\hat{\rho}_t^*$, exists [10, §3.10 – §3.11]. Also by the monotonicity property of $\hat{f}(\lambda; \rho)$, the noise threshold for the recursion (2.19) exists as well. We define the potential function and the potential threshold for the recursion (2.18), respectively, by

$$\begin{aligned} V(\lambda; \rho) &\triangleq \int_0^{\lambda} \left(z - \hat{f}\left(z; \rho\right) \right) \hat{f}'\left(z; \rho\right) \mathrm{d}z \\ &= \int_0^{\lambda} \left(z - \rho\phi(z; t-1) \right) \phi'(z; t-1) \mathrm{d}z, \end{aligned}$$

and

$$\hat{\rho}_t^{**} \triangleq \sup \left\{ \rho \in [0, \infty) \,\middle|\, \min_{\lambda \geq 0} V\left(\lambda; \rho\right) \geq 0 \right\}. \tag{2.31}$$

By [19] as well, the potential threshold $\hat{\rho}_t^{**}$ can be achieved by applying to the $(\mathcal{C}, m, L, w)$ spatially-coupled ensemble in the limit where $m \gg L \gg w$ as $w \to \infty$.

Corollary 22. *For the high-rate scaling limit recursion (2.18), the potential threshold in terms of the average number of errors in a code constraint satisfies $\hat{\rho}_t^{**} \geq 2t - 2$ for all $t \geq t_0$.*

Proof. Consider the DE recursion (2.4) for the $(\mathcal{C}, m)$ GLDPC ensemble. Let $\hat{p}_n^{**}$ be

the potential threshold as defined in (2.29). Note that $\hat{p}_n^{**}$ can be achieved by the $(\mathcal{C}, m, L, w)$ spatially-coupled ensemble in the limit where $m \gg L \gg w$ as $w \to \infty$. For any fixed $n \geq n_0$ and $t \geq t_0$, we define $\hat{\rho}_{n,t}^{**} \triangleq (n-1)\hat{p}_n^{**}$. From Lemma 20, we know that $\hat{\rho}_{n,t}^{**} \geq (2t-2)\frac{n-1}{n}$ for all $n \geq n_0$ and $t \geq t_0$. Thus, we conclude that $\hat{\rho}_t^{**} \triangleq \lim_{n\to\infty} \hat{\rho}_{n,t}^{**} \geq 2t-2$ for all $t \geq t_0$.

2.4.2 Iterative HDD with BDD

For the case of iterative HDD algorithm, it is not clear if the recursion (2.10) defines an admissible system for every linear code (e.g., monotonicity could fail). However, we believe this is the case. Fortunately, one can show that in the high-rate scaling limit, the function $f(\lambda;\rho)$ in (2.22) is strictly increasing in both arguments for $\lambda, \rho > 0$. Therefore, the noise threshold and the potential threshold for the recursion (2.22), denoted by ρ^* and ρ^{**} respectively, exist as well. The potential function and the potential threshold for the recursion (2.22) are, respectively,

$$U(\lambda;\rho) \triangleq \int_0^{\lambda} \left(z - f\left(z;\rho\right)\right) f'\left(z;\rho\right) \mathrm{d}z,$$

and

$$\rho_t^{**} \triangleq \sup\left\{\rho \in [0,\infty) \,\middle|\, \min_{\lambda \geq 0} U\left(\lambda;\rho\right) \geq 0\right\}.$$

Using the threshold of iterative HDD without miscorrection, $\hat{\rho}_t^{**}$, one can obtain a lower bound of ρ_t^{**} from the following lemma.

Lemma 23. *For the high-rate scaling limit, the potential threshold of iterative HDD with miscorrection, ρ_t^{**}, satisfies $\rho_t^{**} \geq \hat{\rho}_t^{**} - \frac{1}{(t-1)!}$.*

Proof. To show the lower bound of ρ_t^{**}, we introduce the recursion

$$\overline{\lambda}^{(\ell+1)} = \left(\rho + \frac{1}{(t-1)!}\right) \phi\left(\overline{\lambda}^{(\ell)}; t-1\right). \tag{2.32}$$

Let $\overline{\rho} \triangleq \rho + \frac{1}{(t-1)!}$. From (2.22), one can show that

$$\begin{aligned} f\left(\lambda; \rho\right) &\leq \rho\phi(\lambda; t-1) + \frac{1}{(t-1)!}\phi(\lambda; t-1) \\ &= \overline{\rho}\phi(\lambda; t-1), \end{aligned} \tag{2.33}$$

and thus

$$\overline{\lambda}^{(\ell+1)} \geq f\left(\overline{\lambda}^{(\ell)}; \rho\right).$$

Therefore, we know that $\lambda^{(\ell)}$ in the recursion (2.22) satisfies $\lambda^{(\ell)} \leq \overline{\lambda}^{(\ell)}$ when the initial values $\lambda^{(0)} = \overline{\lambda}^{(0)}$.

By rewriting (2.32) as $\overline{\lambda}^{(\ell+1)} = \overline{\rho}\phi(\overline{\lambda}^{(\ell)}; t-1)$, the recursion (2.32) can be considered as a $(\overline{\rho}x, \phi(x; t-1))$ system. The update equation of the spatially-coupled $(\overline{\rho}x, \phi(x; t-1))$ system is

$$\overline{\lambda}_i^{(\ell+1)} = \overline{\rho}\left(\frac{1}{w}\sum_{k=0}^{w-1}\phi\left(\frac{1}{w}\sum_{j=0}^{w-1}\overline{\lambda}_{i-j+k}^{(\ell)}; t-1\right)\right), \tag{2.34}$$

where $\overline{\lambda}_i^{(\ell)}$ is the average number of error messages emitted by bit nodes at position

i in the ℓ-th iteration and $i \in \{1, 2, \ldots, L\}$. From (2.33), we know

$$\begin{aligned}\overline{\lambda}_i^{(\ell+1)} &= \frac{1}{w}\sum_{k=0}^{w-1} \overline{\rho}\phi\left(\frac{1}{w}\sum_{j=0}^{w-1}\overline{\lambda}_{i-j+k}^{(\ell)}; t-1\right) \\ &\geq \frac{1}{w}\sum_{k=0}^{w-1} f\left(\frac{1}{w}\sum_{j=0}^{w-1}\overline{\lambda}_{i-j+k}^{(\ell)}; \rho\right).\end{aligned}$$

With the same w and the same initial value, $\lambda_i^{(0)} = \overline{\lambda}_i^{(0)}$ for $i \in \{1, 2, \ldots, L\}$, $\lambda_i^{(\ell)}$ in (2.23) is upper bounded by $\overline{\lambda}_i^{(\ell)}$ for $i \in \{1, 2, \ldots, L\}$ and $\ell \geq 0$.

Denote the potential threshold for the recursion (2.32) by $\overline{\rho}^{**}$. We know that $\overline{\rho}^{**} = \hat{\rho}^{**}$, where $\hat{\rho}^{**}$ is defined in (2.32). According to [19, Theorem 1], for each $\rho < \overline{\rho}^{**} - \frac{1}{(t-1)!} = \hat{\rho}_t^{**} - \frac{1}{(t-1)!}$, there exists a $w_\rho > 0$ such that all $w > w_\rho$, $\overline{\lambda}_i^{(\ell)} \to 0$ as $\ell \to \infty$ for all $i \in \{1, 2, \ldots, L\}$. By the fact that $\lambda_i^{(\ell)} \leq \overline{\lambda}_i^{(\ell)}$ for all $i \in \{1, 2, \ldots, L\}$, we know that $\lambda_i^{(\ell)} \to 0$ as well. This implies that $\rho_t^{**} \geq \hat{\rho}_t^{**} - \frac{1}{(t-1)!}$.

2.5 Approaching Capacity

In this section, we show that the proposed iterative HDD for the spatially-coupled ensemble can approach the capacity in the high-rate regime. We first introduce the notion of ϵ-redundancy.

Definition 24. Let $C(p)$ be the capacity of a BSC(p). For an $\epsilon > 0$, a code ensemble with rate R and threshold p^* is called *ϵ-redundancy achieving* if

$$\frac{1 - C(p^*)}{1 - R} \geq 1 - \epsilon.$$

Let $n_\nu \triangleq 2^\nu - 1$. The following lemma shows that, for any $\epsilon > 0$, a sequence of ensembles with rate $R_\nu = 1 - \frac{2\nu t}{n_\nu}$ and threshold $p_\nu^* = \frac{2t}{n_\nu}$ is ϵ-redundancy achieving over BSC channels when $\nu \in \mathbb{N}$ is large. That is, for any $\epsilon > 0$, there exists a $\nu_0 \in \mathbb{N}$

such that, for all $\nu \geq \nu_0$, one has

$$\frac{1 - C(2tn_\nu^{-1})}{2t\nu n_\nu^{-1}} \geq 1 - \epsilon.$$

Lemma 25. *Consider a sequence of BSCs with error probability $2tn_\nu^{-1}$ for a fixed t and $\nu \in \mathbb{N}$. Then, the ratio of $1 - C\left(2tn_\nu^{-1}\right)$ to $2t\nu n_\nu^{-1}$ goes to 1 as $\nu \to \infty$. That is,*

$$\lim_{\nu \to \infty} \frac{1 - C\left(2tn_\nu^{-1}\right)}{2t\nu n_\nu^{-1}} = 1. \tag{2.35}$$

Proof. Recall that the capacity of the BSC(p) is $1 - H\left(p\right)$, where $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function. The numerator of the LHS of (2.35) can be written as

$$H\left(\frac{2t}{n_\nu}\right) = \frac{2t\log_2 n_\nu}{n_\nu}\left(1 - \frac{\log_2\left(\frac{2t}{e}\right)}{\log_2 n_\nu} - O\left(n_\nu^{-1}\right)\right). \tag{2.36}$$

By substituting (2.36) into the LHS of (2.35), we have

$$\frac{1 - C\left(2tn_\nu^{-1}\right)}{2t\nu n_\nu^{-1}} = \frac{2tn_\nu^{-1}\log_2\left(n_\nu\right)}{2t\nu n_\nu^{-1}}\left(1 - O\left(\nu^{-1}\right)\right).$$

Then, the equality (2.35) follows since $\log_2(n_\nu) = \nu + o(1)$.

With $(n_\nu, k, 2t+1)$ binary primitive BCH codes, the sequence of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC codes can have rate $R \geq 1 - 2tn_\nu^{-1}$. In the following discussion, we will show that the noise threshold of the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble using iterative HDD with BDD, denoted by $p_{n_\nu}^{**}$ and defined in (2.37), satisfies $(n_\nu - 1)p_{n_\nu}^{**} \geq 2t - 2 - \frac{1}{(t-1)!} - \epsilon$ for some $t > 0$, $\epsilon > 0$ and $L \gg w > 0$. Then, we show that the $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC code is ϵ-redundancy achieving when $n \gg t \gg 1$ in Theorem 29.

Lemma 26. *For any $0 \leq \lambda < \infty$ and $0 < t < \infty$, let n satisfy $\lfloor \sqrt{n} \rfloor > \max\{e\lambda, \lambda + 1 + t\}$, and let $X_n \sim B(n-1, \frac{\lambda}{n-1})$ be a binomial random variable with the mean λ. Then,*

$$E\left[nQ_n(X_n)\right] \leq \left(\frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}(t+1, n-t-1).$$

Proof. See Appendix A.6.

Lemma 27. *Given a pair (t, ρ) with $t > 0$ and $\rho \geq 2$, and for each $\epsilon > 0$, there exists a $n_1 > 0$ such that, for all $n \geq n_1$,*

$$(n-1) f_n\left(\frac{\lambda}{n-1}; \frac{\rho}{n-1}\right) \leq \left(\rho + \frac{1}{(t-1)!} + \epsilon\right) \phi\left(\lambda; t-1\right),$$

for all $0 \leq \lambda \leq 3\rho$, where $f_n(x; p)$ is defined in (2.11).

Proof. From (2.21), we know

$$\begin{aligned}
&(n-1) f_n\left(\frac{\lambda}{n-1}; \frac{\rho}{n-1}\right) \\
&\quad \leq \rho I_{\frac{\lambda}{n-1}}\left(t, n-t\right) + \sum_{i=t+1}^{n-1} \binom{n-1}{i} \left(\frac{\lambda}{n-1}\right)^i \left(1 - \frac{\lambda}{n-1}\right)^{n-1-i} nQ_n(i) \\
&\quad \overset{\text{(a)}}{\leq} \rho I_{\frac{\lambda}{n-1}}\left(t, n-t\right) + \left(\frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}(t+1, n-t-1) \\
&\quad \leq \left(\rho + \frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}\left(t, n-t\right).
\end{aligned}$$

where the inequality (a) is obtained by applying Lemma 26. From [54], we know that $I_{\frac{\lambda}{n}}(t, n-t+1)$ converges to $\phi(t-1, \lambda)$ uniformly for $\lambda \in [0, 3\rho]$. For any $\epsilon > 0$, there exists a $n_1 > 0$ such that $I_{\frac{\lambda}{n-1}}\left(t, n-t\right) \leq \phi(\lambda; t-1) + \frac{\epsilon}{2}(\rho + \frac{1}{(t-1)!} + \epsilon)^{-1} \phi(\lambda; t-1)$

for all $0 \leq \lambda \leq 3\rho$, and $O(n^{-0.1}) \leq \frac{\epsilon}{2}$. Then, whenever $n \geq n_1$,

$$\begin{aligned}
&(n-1) f_n\left(\frac{\lambda}{n-1}; \frac{\rho}{n-1}\right) \\
&\quad \leq \left(\rho + \frac{1}{(t-1)!} + \frac{\epsilon}{2}\right)\left(1 + \frac{\epsilon}{2}\left(\rho + \frac{1}{(t-1)!} + \epsilon\right)^{-1}\right)\phi(\lambda; t-1) \\
&\quad \leq \left(\rho + \frac{1}{(t-1)!} + \epsilon\right)\phi(\lambda; t-1).
\end{aligned}$$

For the spatially-coupled $(\mathcal{C}, m, L, w)$ GLDPC codes with iterative BDD, let $\boldsymbol{x}^{(\ell)}(\boldsymbol{x}^{(0)}; p) \in [0,1]^L$ be the vector of error probabilities after ℓ iterations of (2.16) with the initial error probability vector $\boldsymbol{x}^{(0)} \in [0,1]^L$, where the i-th element, denoted by $x_i^{(\ell)}(\boldsymbol{x}^{(0)}; p)$, is the average error probability of the messages emitted by bit nodes at the i-th position. We define the noise threshold for the recursion (2.16) by

$$p_n^{**} \triangleq \sup\left\{p \in (0,1] \,\middle|\, \lim_{\ell \to 0} \boldsymbol{x}^{(\ell)}(p'\mathbf{1}; p') = \mathbf{0} \text{ for all } p' \in [0,p]\right\}. \tag{2.37}$$

Corollary 28. *Given a pair (t, ρ) with $t > 0$ and $\rho \geq 2$, and for each $\epsilon > 0$, there exists a $n_1 > 0$ such that, whenever $n_1 \leq n < \infty$, the noise threshold p_n^{**} satisfies $(n-1)p_n^{**} \geq 2t - 2 - \frac{1}{(t-1)!} - \epsilon$ when $m \gg L \gg w$ and $w \to \infty$.*

Proof. According to Lemma 27, we consider the following recursion

$$\overline{\lambda}^{(\ell+1)} = \overline{\rho}\phi(\overline{\lambda}^{(\ell)}; t-1), \tag{2.38}$$

where

$$\overline{\rho} \triangleq (n-1)p + \frac{1}{(t-1)!} + \epsilon. \tag{2.39}$$

Since the recursion (2.38) can also be considered as a $(\overline{\rho}x, \phi(x; t-1))$ admissible system, and the DE update equation of the spatially-coupled system is shown in

(2.34). For each $\epsilon > 0$, let n_1 be selected according to the proof of Lemma 27. Then, we know that, when $n > n_1$,

$$\begin{aligned}
\frac{\overline{\lambda}_i^{(\ell+1)}}{n-1} &= \frac{1}{n-1}\overline{\rho}\left(\frac{1}{w}\sum_{k=0}^{w-1}\phi\left(\frac{1}{w}\sum_{j=0}^{w-1}\overline{\lambda}_{i-j+k}^{(\ell)}; t-1\right)\right) \\
&\geq \frac{1}{n-1}\left(\frac{1}{w}\sum_{k=0}^{w-1}(n-1)f_n\left(\frac{1}{w}\sum_{j=0}^{w-1}\frac{\overline{\lambda}_{i-j+k}^{(\ell)}}{n-1}; \frac{\rho}{n-1}\right)\right) \\
&= \frac{1}{w}\sum_{k=0}^{w-1} f_n\left(\frac{1}{w}\sum_{j=0}^{w-1}\frac{\overline{\lambda}_{i-j+k}^{(\ell)}}{n-1}; \frac{\rho}{n-1}\right). \qquad (2.40)
\end{aligned}$$

Note that the RHS of the last equality in (2.40) is the update equation (2.16) with $x_i^{(\ell)} = \frac{\overline{\lambda}_i^{(\ell)}}{n-1}$ and $p = \frac{\rho}{n-1}$. When $(n-1)x_i^{(0)} = \overline{\lambda}_i^{(0)} = \lambda$ for all $i \in \{1, 2, \ldots, L\}$ and $p = \frac{\rho}{n-1}$, one can show that $\frac{\overline{\lambda}_i^{(\ell)}}{n-1} \geq x_i^{(\ell)}$ for all $\ell \geq 0$ by induction.

Let $\overline{\rho}^{**}$ be the potential threshold of the recursion (2.38). For each $\overline{\rho} < \overline{\rho}^{**}$, there exists a $w_{\overline{\rho}} > 0$ such that for all $w > w_{\overline{\rho}}$, $\overline{\lambda}_i^{(\ell)} \to 0$ for all $i \in \{1, 2, \ldots, L\}$. Since $\frac{\overline{\lambda}_i^{(\ell)}}{n-1} \geq x_i^{(\ell)}$, we also know that $x_i^{(\ell)} \to 0$ whenever p satisfies

$$(n-1)p + \frac{1}{(t-1)!} + \epsilon < \overline{\rho}^{**}.$$

Thus the potential threshold p_n^{**} is lower bounded by

$$p_n^{**} \geq \frac{1}{n-1}\left(\overline{\rho}^{**} - \frac{1}{(t-1)!} - \epsilon\right).$$

Since $\overline{\rho}^{**} = \hat{\rho}^{**} \geq 2t-2$, we conclude that $(n-1)p_n^{**} \geq 2t - 2 - \frac{1}{(t-1)!} - \epsilon$.

The following theorem shows that iterative HDD of the spatially-coupled GLDPC ensemble approaches capacity in high-rate regime.

Theorem 29. *For any $\epsilon > 0$, there exists a tuple (t, n, L, w) such that iterative HDD of the $(\mathcal{C}, m, L, w)$ GLDPC spatially-coupled ensemble is ϵ-redundancy achieving when*

$\mathcal{C}$ is a t-error correcting BCH code of length n.

Proof. We prove the theorem by showing the existence of the tuple (t, n, L, w) such that for a given $\epsilon \in (0, 1)$, the $(\mathcal{C}, \infty, L, w)$ GLDPC spatially-coupled ensemble with the proposed iterative HDD algorithm is ϵ-redundancy achieving.

We first let $t \geq \max\{\frac{8}{\epsilon}, t_0\}$, where t_0 is defined in Lemma 20. Select a $\nu_1 > 0$ and define $n_{\nu_1} = 2^{\nu_1} - 1$ such that $H(\frac{2t}{n_{\nu_1}}) \geq 2t\nu n_{\nu_1}^{-1}(1 - \frac{\epsilon}{4}) \log_2 n_{\nu_1}$. From the threshold of the high-rate scaling limit introduced in Lemma 23, we know that the noise threshold of the spatially-coupled system is around $2t$. Thus, we can consider the channel noise in terms of the average number of errors per code $\rho \in [0, 3t]$. By Lemma 27, there exists a $n_1 > 0$ such that $(n-1) f_n\left(\frac{\lambda}{n-1}; \frac{\rho}{n-1}\right) \leq \left(\rho + \frac{1}{(t-1)!} + \frac{1}{2}\right) \phi(\lambda; t-1)$ for all $0 \leq \lambda \leq 3\rho$. Let $\nu \geq \lceil \max\{\log_2 n_t, \log_2 n_0, \log_2 n_1, \nu_1\} \rceil$ and $n = 2^\nu - 1$. From Corollary 28, we know the noise threshold of the spatially-coupled recursion of (2.21) satisfies $(n-1)p^{**} \geq 2t - 2 - \frac{1}{(t-1)!} - \frac{1}{2}$. By selecting $p = (2t-4)(n-1)^{-1}$, we know that there exists a $0 < w < \infty$ such that the spatially-coupled recursion of (2.21) converges to 0 as $\ell \to \infty$. After determining w, we select L such that $L \geq 2(w-1)\epsilon^{-1}$. Thus, the rate loss due to spatial coupling (2.1) is $R \geq 1 - \frac{2t\nu}{n_\nu}(1 + \frac{\epsilon}{2})$. Finally, we conclude the proof by showing that the $(\mathcal{C}, m, L, w)$ spatially-coupled ensemble is ϵ-redundancy achieving by

$$\frac{1 - C(p)}{1 - R} \geq \frac{(2t-4)\,\nu n_\nu^{-1}\left(1 - \frac{\epsilon}{4}\right)}{2t\nu n_\nu^{-1}\left(1 + \frac{\epsilon}{2}\right)} = \frac{(2t-4)\left(1 - \frac{\epsilon}{4}\right)}{2t\left(1 + \frac{\epsilon}{2}\right)} \geq \frac{\left(1 - \frac{\epsilon}{4}\right)^2}{\left(1 + \frac{\epsilon}{2}\right)} \geq 1 - \epsilon.$$

2.6 Practical Implementation of Iterative HDD

In this section, we describe the practical implementation of the iterative HDD described in Section 2.2. We highlight the difference between conventional decoding, which we call intrinsic message passing (IMP), and the proposed approach in Section

2.2, which we call extrinsic message passing (EMP). In EMP algorithms, messages passed on edges in the Tanner graph are computed only from their extrinsic information. For certain random ensembles, this enables analysis via density evolution. We emphasize that this is different than the conventional iterative HDD rule typically used by product codes. In contrast to Section 2.2.4, this section introduces the EMP algorithm in a message-passing fashion to make it clear that the EMP uses only the extrinsic information.

Let $\boldsymbol{r}$ be the vector of channel output bits, $\nu_{i,j}^{(\ell)} \in \{0,1\}$ be the messages passed from the i-th bit node to the j-th constraint node in the ℓ-th iteration, and $\mu_{i,j}^{(\ell)} \in \{0,1\}$ be the messages passed from the j-th constraint node to the i-th bit node in the ℓ-th iteration. We assume the constraint nodes define an $(n, k, d_{\min})$ component code $\mathcal{C}$ with $d_{\min} = 2t+1$. Let $\boldsymbol{v} \in \{0,1\}^n$ be an input vector to a constraint node, and let $\mathsf{D} : \{0,1\}^n \to \mathcal{C} \cup \{\mathtt{fail}\}$ be the operator of bounded distance decoding (BDD) with decoding radius t defined by

$$\mathsf{D}(\boldsymbol{v}) = \begin{cases} \boldsymbol{c} & \text{if } \boldsymbol{c} \in \mathcal{C} \text{ and } d_H(\boldsymbol{v}, \boldsymbol{c}) \leq t \\ \mathtt{fail} & \text{otherwise.} \end{cases}$$

2.6.1 Intrinsic Message Passing

In this section, we recall the IMP algorithm for highlighting the difference with EMP. For a bit node i and a constraint node j, let $\mathcal{N}(i) = \{j, j'\}$ be the constraint-node neighbors of i. Let $\boldsymbol{\nu}_j^{(\ell)} \triangleq (\nu_{\sigma_j(1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(n),j}^{(\ell)})$ be the collection of the all input messages to the j-th constraint node in the ℓ-th iteration, where $\sigma_j(k) \in \mathcal{I}$ is defined in Section 2.2. Let $\boldsymbol{w}_j^{(\ell)} \triangleq (w_{1,j}^{(\ell)}, \ldots, w_{n,j}^{(\ell)}) = \mathsf{D}(\boldsymbol{\nu}_j^{(\ell)})$ be the output of the BDD decoder applied to the input messages at the j-th constraint node. Then, the

message-passing rules, for each constraint node j, are

$$\nu_{i,j}^{(\ell+1)} = \mu_{i,j'}^{(\ell)} \tag{2.41}$$

$$\mu_{\sigma_j(k),j}^{(\ell)} = \begin{cases} w_{k,j}^{(\ell)} & \text{if } \mathsf{D}\big(\boldsymbol{\nu}_j^{(\ell)}\big) \neq \texttt{fail} \\ \nu_{\sigma_j(k),j}^{(\ell)} & \text{otherwise.} \end{cases} \tag{2.42}$$

The iteration starts by initializing $\nu_{i,j}^{(0)} = r_i$ for each bit node i and all $j \in N(i)$. From (2.41) and (2.42), one can see that the IMP algorithm only uses channel outputs at the beginning of the iterations, and then, exchanges the output of the constraint nodes in the subsequent iterations. The IMP is the conventional approach used for the iterative HDD of product codes.

2.6.2 Extrinsic Message Passing

In the IMP message-passing rule (2.42), the computation of the output message $\mu_{i,j}^{(\ell)}$ passed from j to i uses the input message $\nu_{i,j}^{(\ell)}$. This violates the principle of using only extrinsic information in message-passing rules. The decoding algorithm proposed in Section 2.2 can rectify this problem. We note that the messages in the EMP decoder are denoted by $\hat{\nu}_{i,j}^{(\ell)}$ and $\hat{\mu}_{i,j}^{(\ell)}$ to distinguish them from the IMP decoder.

Let $\hat{\nu}_{i,j}^{(\ell)} \in \{0,1\}$ be the message passed by the EMP algorithm from the i-th bit node to the j-th constraint node and let $\hat{\boldsymbol{\nu}}_j^{(\ell)} \triangleq (\hat{\nu}_{\sigma_j(1),j}^{(\ell)}, \ldots, \hat{\nu}_{\sigma_j(n),j}^{(\ell)})$ be the collection of all input messages to the j-th constraint node in the ℓ-th iteration. To compute the EMP message $\hat{\mu}_{i,j}^{(\ell)} \triangleq \big(\hat{\mu}_{i,j,0}^{(\ell)}, \hat{\mu}_{i,j,1}^{(\ell)}\big)$ from the j-th constraint node to the i-th bit node, BDD is performed twice. In the ℓ-th iteration, similar to the arrangement of the candidate decoding vector in (2.3), we first define

$$\hat{\boldsymbol{\nu}}_{j,k,0}^{(\ell)} \triangleq (\hat{\nu}_{\sigma_j(1),j}^{(\ell)}, \ldots, \hat{\nu}_{\sigma_j(k-1),j}^{(\ell)}, 0, \hat{\nu}_{\sigma_j(k+1),j}^{(\ell)}, \ldots, \hat{\nu}_{\sigma_j(n),j}^{(\ell)})$$

and

$$\hat{\boldsymbol{\nu}}_{j,k,1}^{(\ell)} \triangleq (\hat{\nu}_{\sigma_j(1),j}^{(\ell)}, \ldots, \hat{\nu}_{\sigma_j(k-1),j}^{(\ell)}, 1, \hat{\nu}_{\sigma_j(k+1),j}^{(\ell)}, \ldots, \hat{\nu}_{\sigma_j(n),j}^{(\ell)}),$$

and then computes $\hat{\boldsymbol{w}}_{j,k,0}^{(\ell)} = \mathsf{D}(\hat{\boldsymbol{\nu}}_{j,k,0}^{(\ell)})$ and $\hat{\boldsymbol{w}}_{j,k,1}^{(\ell)} = \mathsf{D}(\hat{\boldsymbol{\nu}}_{j,k,1}^{(\ell)})$, respectively. Based on $\hat{\boldsymbol{w}}_{j,k,0}^{(\ell)}$ and $\hat{\boldsymbol{w}}_{j,k,1}^{(\ell)}$ computed at the j-th constraint node, the messages $\hat{\mu}_{\sigma_j(k),j,0}^{(\ell)}$ and $\hat{\mu}_{\sigma_j(k),j,1}^{(\ell)}$ is assigned, respectively, by

$$\hat{\mu}_{\sigma_j(k),j,0}^{(\ell)} = \begin{cases} \left[\hat{\boldsymbol{w}}_{j,k,0}^{(\ell)}\right]_k & \text{if } \mathsf{D}\big(\hat{\boldsymbol{\nu}}_{j,k,0}^{(\ell)}\big) \neq \texttt{fail} \\ \texttt{fail} & \text{otherwise,} \end{cases}$$

and

$$\hat{\mu}_{\sigma_j(k),j,1}^{(\ell)} = \begin{cases} \left[\hat{\boldsymbol{w}}_{j,k,1}^{(\ell)}\right]_k & \text{if } \mathsf{D}\big(\hat{\boldsymbol{\nu}}_{j,k,1}^{(\ell)}\big) \neq \texttt{fail} \\ \texttt{fail} & \text{otherwise.} \end{cases}$$

One can see that the message $\hat{\mu}_{i,j}^{(\ell)}$ will be in the set

$$\{(0,0), (1,1), (0,1), (0,\texttt{fail}), (\texttt{fail},1), (\texttt{fail},\texttt{fail})\}.$$

We recall that $\mathcal{N}(i) = \{j, j'\}$. The message-passing rule for the i-th bit node is

$$\hat{\nu}_{i,j}^{(\ell+1)} \triangleq \begin{cases} 0 & \text{if } \hat{\mu}_{i,j'}^{(\ell)} = (0,0) \\ 1 & \text{if } \hat{\mu}_{i,j'}^{(\ell)} = (1,1) \\ r_i & \text{otherwise.} \end{cases} \tag{2.43}$$

The iteration is initialized by setting $\hat{\nu}_{i,j}^{(0)} = r_i$ for each bit node i and all $j \in \mathcal{N}(i)$.

By replacing the k-th element of $\hat{\boldsymbol{\nu}}_j^{(\ell)}$ with both 0 and 1, the computed output $\hat{\mu}_{\sigma_j(k),j}^{(\ell)}$ remains independent of the incoming message $\hat{\nu}_{\sigma_j(k),j}^{(\ell)}$ on that edge. Therefore,

only extrinsic information is used in the computation of the output message on the $(\sigma_j(k), j)$ edge from the j-th constraint node. The output message from a bit node depends only on the channel observation and the input from the other edge. Therefore, this defines an extrinsic message-passing algorithm with hard-decision messages.

2.6.3 Low-Complexity EMP Algorithm

As described above, the EMP algorithm needs to run the BDD algorithm $2n$ times to compute the output messages from a single constraint node. The primary purpose of that description was to demonstrate that the algorithm is indeed an EMP algorithm. Now, we show that exactly the same outputs can be computed with a single decode and some post processing. In the ℓ-th iteration, let $\boldsymbol{w} \triangleq \mathsf{D}(\hat{\boldsymbol{\nu}}_j^{(\ell)})$ be the output of the BDD at the j-th constraint node with $\hat{\boldsymbol{\nu}}_j^{(\ell)}$ as an input. In this case, we will see that one can calculate $\hat{\boldsymbol{\nu}}_j^{(\ell+1)}$ directly from $\hat{\boldsymbol{\nu}}_j^{(\ell)}$. In this section, the $\hat{\mu}_{\sigma_j(k),j}^{(\ell)}$ messages are used only to explain the correctness of the simplified algorithm. Consider the following facts.

Proof 30. If $\boldsymbol{w} = \texttt{fail}$, then at least one element of $\hat{\mu}_{\sigma_j(k),j}^{(\ell)}$ will be a `fail` for all $k = 1, 2, \ldots, n$. By (2.43), one can show that $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = r_{\sigma_j(k)}$ for all $k = 1, 2, \ldots, n$.

Proof 31. If $\boldsymbol{w} \neq \texttt{fail}$ and $d_H(\hat{\boldsymbol{\nu}}_j^{(\ell)}, \boldsymbol{w}) < t$, then one can show that $\hat{\mu}_{\sigma_j(k),j,0}^{(\ell)} = \hat{\mu}_{\sigma_j(k),j,1}^{(\ell)}$ for all $k = 1, 2, \ldots, n$. Thus, we have $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = \hat{\mu}_{\sigma_j(k),j,0}^{(\ell)}$ for all $k = 1, 2, \ldots, n$.

Proof 32. If $\boldsymbol{w} \neq \texttt{fail}$ and $d_H(\hat{\boldsymbol{\nu}}_j^{(\ell)}, \boldsymbol{w}) = t$, then first suppose that $w_k = \hat{\nu}_{\sigma_j(k),j}^{(\ell)}$ for some k. One can see that $\hat{\mu}_{\sigma_j(k),j}^{(\ell)}$ must not be $(0, 0)$ or $(1, 1)$. Thus, we know $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = r_{\sigma_j(k)}$. On the other hand, suppose that $w_k \neq \hat{\nu}_{\sigma_j(k),j}^{(\ell)}$. One can show that $\hat{\mu}_{\sigma_j(k),j,0}^{(\ell)} = \hat{\mu}_{\sigma_j(k),j,1}^{(\ell)} = w_k$. Therefore, $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = w_k$.

Using these facts, we define the low-complexity EMP decoder in Algorithm 1. Since the distance $d_H(\hat{\boldsymbol{\nu}}_j^{(\ell)}, \boldsymbol{w})$ is automatically obtained while performing BDD, the only overhead of the EMP algorithm is a little Boolean logic.

Algorithm 1 The low-complexity EMP algorithm

Iteration ℓ: For each constraint node j,

- Compute $\boldsymbol{w} = \mathsf{D}(\hat{\boldsymbol{\nu}}_j^{(\ell)})$.
- For $k = 1, \ldots, n$,
  - if $d_H(\hat{\boldsymbol{\nu}}_j^{(\ell)}, \boldsymbol{w}) > t$, then $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = r_{i_k}$
  - elseif $d_H(\hat{\boldsymbol{\nu}}_j^{(\ell)}, \boldsymbol{w}) < t$, then $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = w_k$
  - else $\hat{\nu}_{\sigma_j(k),j'}^{(\ell+1)} = \big(\big(1 - \hat{\nu}_{\sigma_j(k),j}\big)\big(r_{\sigma_j(k)} \,\mathtt{OR}\, w_k\big)\big) \,\mathtt{OR}\, (r_{\sigma_j(k)} w_k)$.

Remark 33. While preparing this extended version of our earlier work [53], we notice that Miladinovic and Fossorier also proposed an iterative HDD algorithm for general product codes [13]. We briefly describe their algorithm as follows. For an edge (i, j) connecting the bit node i and the constraint node j, let $j' = \mathcal{N}(i) \setminus j$, $i = \sigma_j(k)$, and $\boldsymbol{\nu}_j^{(\ell)} \triangleq (\nu_{\sigma_j(1),j}^{(\ell)}, \nu_{\sigma_j(2),j}^{(\ell)} \ldots, \nu_{\sigma_j(n),j}^{(\ell)})$. Note that the k-th element of $\boldsymbol{\nu}_j^{(\ell)}$ is $\nu_{i,j}^{(\ell)}$. The message passed by the constraint node j, denoted by $\mu_{i,j}^{(\ell)}$, consists of two elements $\mu_{i,j}^{(\ell)} \triangleq (\mathsf{D}_k(\boldsymbol{\nu}_j^{(\ell)}), s^{(\ell)})$, where $s^{(\ell)} = 1$ if the decoding at the j-th constraint node has succeeded; otherwise, $s^{(\ell)} = 0$. At the i-th bit node, the message $\nu_{i,j'}^{(\ell+1)}$ is updated by

$$\nu_{i,j'}^{(\ell+1)} = \big(1 - s^{(\ell)}\big)\, r_i + s \mathsf{D}_k \left(\boldsymbol{\nu}_j^{(\ell)}\right). \tag{2.44}$$

One can see that the proposed algorithm is similar to the iterative HDD algorithm proposed. However, the outputs of the two iterative HDD algorithm are different when $t = \frac{d_{\min}-1}{2}$, $\boldsymbol{c} = \mathsf{D}(\boldsymbol{\nu}_j^{(\ell)})$, $d_H(\boldsymbol{c}, \boldsymbol{\nu}_j^{(\ell)}) = t$, and $c_k = \nu_{i,j}^{(\ell)} \neq r_i$. For the proposed iterative HDD and the vector $\boldsymbol{v}_{i,j}^{(\ell)}$ defined in (2.3), we know that $\mathsf{D}(\boldsymbol{v}_{i,j}^{(\ell)})$ will be r_i, but $(1-s)r_i + s\mathsf{D}_k(\boldsymbol{\nu}_j^{(\ell)}) = c_k$. Moreover, we notice that $\nu_{i,j'}^{(\ell+1)}$ in the update equation (2.44) will depend on $\mu_{i,j'}^{(\ell-1)}$. In the ℓ-th iteration, we define two vectors

$$\boldsymbol{\nu}_{j,k,0}^{(\ell)} \triangleq \left(\nu_{\sigma_j(1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(k-1),j}^{(\ell)}, 0, \nu_{\sigma_j(k+1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(n),j}^{(\ell)}\right) \tag{2.45}$$

and

$$\boldsymbol{\nu}_{j,k,1}^{(\ell)} \triangleq \left(\nu_{\sigma_j(1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(k-1),j}^{(\ell)}, 1, \nu_{\sigma_j(k+1),j}^{(\ell)}, \ldots, \nu_{\sigma_j(n),j}^{(\ell)}\right). \tag{2.46}$$

Suppose that the decoder outputs $\boldsymbol{c} = \mathsf{D}(\boldsymbol{\nu}_{j,k,0}^{(\ell)})$ and $\boldsymbol{c}' = \mathsf{D}(\boldsymbol{\nu}_{j,k,1}^{(\ell)})$ are both codewords. It is clear that $d_H(\boldsymbol{c}, \boldsymbol{\nu}_{j,k,0}^{(\ell)}) = d_H(\boldsymbol{c}', \boldsymbol{\nu}_{j,k,1}^{(\ell)}) = t$, $\boldsymbol{c}_k = 0$, and $\boldsymbol{c}'_k = 1$. Also, we know $s = 1$ for the decoding of both vectors. The possible values of $\nu_{i,j'}^{(\ell+1)}$ are listed in Table 2.1. Since the values of $\nu_{i,j'}^{(\ell+1)}$ for different $\mu_{i,j'}^{(\ell-1)}$ are different, we observe that $\nu_{i,j'}^{(\ell+1)}$ is not independent of $\mu_{i,j'}^{(\ell-1)}$.

Table 2.1: The possible values of $\nu_{i,j'}^{(\ell+1)}$ with input vectors $\mu_{i,j'}^{(\ell-1)}$ when $\boldsymbol{c} = \mathsf{D}(\boldsymbol{\nu}_{j,k,0}^{(\ell)})$ and $\boldsymbol{c}' = \mathsf{D}(\boldsymbol{\nu}_{j,k,1}^{(\ell)})$ are codewords, where $\boldsymbol{\nu}_{j,k,0}^{(\ell)}$ and $\boldsymbol{\nu}_{j,k,1}^{(\ell)}$ are defined in (2.45) and (2.46), respectively.

$\mu_{i,j'}^{(\ell-1)}$	$\nu_{i,j}^{(\ell)}$	$\mu_{i,j}^{(\ell)}$	$\nu_{i,j'}^{(\ell+1)}$
$(0,1)$	0	$(0,1)$	0
$(1,1)$	1	$(1,1)$	1

2.7 Numerical Results and Comparison

In the following numerical results, the iterative HDD threshold of $(\mathcal{C}, m, L, w)$ spatially-coupled GLDPC ensemble with $L = 1025$, and $w = 16$ are considered. In Table 2.2, the thresholds of the ensembles are shown in terms of the average number of error bits attached to a code-constraint node. Let $p^*_{n,t}$ be the iterative HDD threshold of the spatially-coupled GLDPC ensemble based on a $(n, k, 2t + 1)$ binary primitive BCH component code, and $\tilde{p}^*_{n,t}$ be the iterative HDD threshold of the spatially-coupled GLDPC ensemble based on the $(n, k-1, 2t+2)$ even-weight subcode. Then, we define $a^*_{n,t} \triangleq np^*_{n,t}$ and $\tilde{a}^*_{n,t} \triangleq n\tilde{p}^*_{n,t}$ to be the thresholds in terms of the average number of error bits attached to a component code. In the high-rate scaling limit, we let ρ^*_t and $\tilde{\rho}^*_t$ denote the iterative HDD thresholds of the ensembles based on primitive BCH component codes and their even-weight subcodes, respectively. Moreover, the threshold of HDD without miscorrection, $\hat{\rho}^*_t$, is shown in Table 2.2 along with the potential threshold, $\hat{\rho}^{**}_t$, of iterative HDD without miscorrection from (2.18).

From Table 2.2, one can observe that the thresholds (ρ^*_t, $\tilde{\rho}^*_t$ and $\hat{\rho}^*_t$) of the spatially-coupled ensemble with primitive BCH component codes or the even-weight subcodes approach to $2t$ as t increases. This verifies the results predicted by Lemma 22 and the vanishing impact of miscorrection predicted by Lemma 23.

2.8 Conclusion

The iterative HDD of GLDPC ensembles, based on on t-error correcting block codes, is analyzed with and without spatial coupling. Using DE analysis, noise thresholds are computed for a variety of component codes and decoding assumptions. In particular, the case of binary primitive BCH component-codes is considered along with their even-weight subcodes. For these codes, the miscorrection probability is characterized and included in the DE analysis. Scaled DE recursions are also

computed for the high-rate limit. When miscorrection is neglected, the resulting recursion for the basic ensemble matches the results of [44, 45]. It is also proven that iterative HDD threshold of the spatially-coupled GLDPC ensemble can approach capacity in high-rate regime. Finally, numerical results are presented that both verify the theoretical results and demonstrate the effectiveness of these codes for high-speed communication systems.

Table 2.2: Iterative HDD thresholds of $(\mathcal{C}, m, 1025, 16)$ spatially-coupled GLDPC ensemble with binary primitive BCH codes

t	3	4	5	6	7
$a^*_{255,t}$	5.432	7.701	9.818	11.86	13.87
$a^*_{511,t}$	5.417	7.665	9.811	11.86	13.85
$a^*_{1023,t}$	5.401	7.693	9.821	11.87	13.88
ρ^*_t	5.390	7.688	9.822	11.91	13.93
$\tilde{a}^*_{255,t}$	5.610	7.752	9.843	11.88	13.87
$\tilde{a}^*_{511,t}$	5.570	7.767	9.811	11.86	13.85
$\tilde{a}^*_{1023,t}$	5.606	7.765	9.841	11.88	13.88
$\tilde{\rho}^*_t$	5.605	7.761	9.840	11.91	13.93
$\hat{\rho}^*_t$	5.735	7.813	9.855	11.91	13.93
$\hat{\rho}^{**}_t$	5.754	7.843	9.896	11.93	13.95

3. CONVERGENCE OF WEIGHTED MIN-SUM DECODING VIA DYNAMIC PROGRAMMING ON TREES

3.1 Introduction

The introduction of turbo codes in 1993 started a revolution in coding and inference that continued with the rediscovery of low-density parity-check (LDPC) codes and culminated in optimized LDPC codes that essentially achieve the capacity of practical channels [3, 4, 5, 29]. During this time, Wiberg *et al.* advanced the analysis of iterative decoding by proving a number of results for the min-sum (MS) decoding algorithm, which is equivalent to the max-product (MP) decoding algorithm [55, 56, 57]. Richardson and Urbanke also introduced the technique of density evolution (DE) to compute noise thresholds of message-passing decoding algorithms for turbo and LDPC codes [9].

For a particular noise realization, the optimality of iterative decoding solutions has also been considered by a number of authors. Weiss and Freeman have shown that the max-product assignment is locally optimal w.r.t. all single-loop and tree perturbations [27]. Unfortunately, this result is typically uninformative for LDPC codes with variables degrees larger than 2. Frey and Koetter have also shown that, with proper weights and adjustments, the attenuated max-product (AttMP) decoding algorithm for LDPC codes returns the maximum-likelihood (ML) codeword if the algorithm converges to a codeword [58]. For general graphs, Wainwright *et al.* proposed the tree-reweighted max-product (TRMP) message-passing algorithm which attempts to compute the MAP assignment on strictly positive Markov random fields

[59]; in fact, they have shown that, under some optimality conditions, the converged solution gives the MAP configuration for the graph. Their algorithm, though strictly different, has some similarity to the AttMP algorithm in [58].

Linear programming (LP) decoding for LDPC codes, proposed by Feldman *et al.*, solves a relaxed version of the ML decoding problem [60]. Since its introduction, a number of authors have looked for connections to the MP algorithm decoding [61, 62]. One interesting open question is, "What is the noise threshold of LP decoding?". In [63], Vontobel and Koetter introduced a necessary condition for LP decoding to return the correct codeword under the all-zero codeword assumption. Based on this necessary condition, upper bounds of the noise threshold of LP decoding for various LDPC codes under the binary symmetric channel (BSC) were obtained. The first lower bound of the noise threshold of LP decoding under the BSC was proposed in [64]. Using expander graph arguments, they showed that LP decoding of a rate-$\frac{1}{2}$ regular LDPC code can correct all error patterns with weight less than 0.000175 of the block length. Since this is a worst-case analysis, the large gap to the empirical observations is not too surprising. Daskalakis *et al.* [65] were able to improve the lower bound to 0.002 using probabilistic arguments based on a construction of a LP dual feasible solution under the BSC. In [66], Koetter and Vontobel applied girth-based arguments to the dual LP problem. For a $(3, 6)$-regular LDPC code, they proved that LP decoding can tolerate a crossover probability of $p = 0.01$ on the BSC and a noise level of $\sigma = 0.5574$ on the binary-input additive white Gaussian noise channel (BIAWGNC).

Arora *et al.* showed recently that, for a $(3, 6)$-regular LDPC code, LP decoding can tolerate a crossover probability $p = 0.05$ on the BSC [1]. Instead of using a dual LP solution, they investigated the primal solution of the LP problem and proposed a local optimality condition for codewords. They proved that the local

optimality implies both global optimality and LP optimality. So the probability that LP decoding succeeds is lower bounded by the probability that the correct codeword satisfies a set of local optimality conditions. Since their local optimality conditions are amenable to analysis on tree-like neighborhoods, they perform a DE-type analysis to obtain the lower bound of the BSC noise thresholds for LP decoding. Using similar DE-type analysis for memoryless binary-input output symmetric (MBIOS) channels, Halabi and Even showed that LP decoding can achieve a noise threshold $\sigma = 0.735$ on the BIAWGNC [67]. By using the same local optimality argument, the authors further extended the sufficient conditions for the ML certificate to the class of GLDPC codes [68, 69].

Similar to the connection between the MP algorithm and the MS algorithm, the AttMP algorithm, for any finite number of iterations, is equivalent to the WMS algorithm proposed by Chen and Fossorier [70]. They showed that, for regular LDPC codes, the WMS algorithm with a proper choice of the weight factor, β, can have the noise threshold better than the noise threshold of the MS algorithm. Therefore, both the AttMP and the WMS algorithms are considered in this section. The results can be seen as an extension of the work by Frey and Koetter that provides new insight into the results of [66, 1]. In [71], Mooij and Kappen characterized sufficient conditions for the convergence of the sum-product algorithm by using the contraction mapping theorem [72, pp. 97–100]. With the same approach, sufficient conditions for the convergence of the AttMP and the WMS algorithms are analyzed. We view both the AttMP and the WMS [70] algorithms as computing the dynamic-programming (DP) solution to the optimal discounted-reward problem on a set of overlapping trees. This allows us to show that, for any received vector, the one-step update of the algorithm is a contraction on the space of message values when the weight factor is sufficiently small. From this, we deduce that the messages converge to a

unique fixed point [72, pp. 97–100]. We first show that, for $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes, if the resulting fixed point satisfies some consistency conditions, then the hard decisions obtained based on the fixed point must be an LP optimum solution and, hence, an ML decoding solution. Then, the WMS algorithm on $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes with messages diverging to $\pm\infty$ is considered. We show that, for the weight $\beta = \frac{1}{d_{\mathrm{v}}-1}$, if the WMS messages diverge to $\pm\infty$ and satisfy the consistency conditions, the corresponding hard decisions also return an ML decoding solution.

The rest of the section is organized as follows. Section 3.2 provides the background on factor graphs as well as the message update rules of the AttMP algorithm and the WMS algorithm. In Section 3.3, we first investigate the convergence property of both algorithms, and then introduce consistency conditions for both algorithms. Then, the ML optimality certificate property is discussed for hard decisions associated with a consistent fixed point. In Section 3.4, the optimality of a codeword returned by the WMS algorithm is analyzed for the case when the message diverges. A conjecture about the connection between noise thresholds of WMS decoding and noise thresholds of LP decoding is proposed in the same section. Numerical results are described and discussed in Section 3.5. Finally, conclusions and extensions are given in Section 3.6.

3.2 Background

3.2.1 Factor Graphs

A binary LDPC code can be defined by a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges, and $\mathcal{V} = \mathcal{V}_{\mathrm{v}} \cup \mathcal{V}_{\mathrm{c}}$ consists of variable nodes (or bit nodes) $\mathcal{V}_{\mathrm{v}}$ and check nodes (or constraint nodes) $\mathcal{V}_{\mathrm{c}}$. In this section, $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes are considered. That is, each variable node in $\mathcal{V}_{\mathrm{v}}$ has d_{v} edges attached to it, and each check node in $\mathcal{V}_{\mathrm{c}}$ has d_{c} edges attached to it. For a set $\mathcal{S}$, let $|\mathcal{S}|$ denote the cardinality of $\mathcal{S}$. Let $n \triangleq |\mathcal{V}_{\mathrm{v}}|$ be the number of variable nodes. Let $N(i)$ be the

neighbors of a node $i \in \mathcal{V}$. For a check node $j \in \mathcal{V}_{\rm c}$, we say that j is satisfied when the binary values assigned to the bit nodes in $N(j)$ satisfy the parity check imposed in j. Any binary vector $\boldsymbol{x} \in \{0,1\}^n$ is a codeword, or a valid assignment, if and only if it satisfies all check nodes in $\mathcal{V}_{\rm c}$. We use $\mathcal{C}$ to denote the collection of all codewords. Let $\mathcal{T}_i^L$ be the computation tree of $\mathcal{G}$ which has depth L and is rooted at node i [55, §4.1]. The set of vertices in the ℓ-th level of $\mathcal{T}_i^L$, where $\ell \leq L$, is denoted by $N(i,\ell)$. The girth of a graph $\mathcal{G}$ is the length of the smallest cycle in $\mathcal{G}$, denoted by girth($\mathcal{G}$).

Suppose that the computation tree $\mathcal{T}_i^{2L}$ has depth $L < \frac{1}{4}\text{girth}(\mathcal{G})$. Then, each node in $\mathcal{T}_i^{2L}$ is associated with a different node in $\mathcal{G}$. Let $\mathcal{I}_{\rm v} \subseteq \mathcal{V}_{\rm v}$ and $\mathcal{I}_{\rm c} \subseteq \mathcal{V}_{\rm c}$ be the subset of variable nodes associated with the variable nodes in $\mathcal{T}_i^{2L}$ and the subset of check nodes associated with the check nodes in $\mathcal{T}_i^{2L}$, respectively. A binary vector $\boldsymbol{w} \in \{0,1\}^n$ is a valid assignment on $\mathcal{T}_i^{2L}$ if $\boldsymbol{w}$ satisfies all check nodes in $\mathcal{I}_{\rm c}$. Let $\mathcal{C}_{\mathcal{T}_i^{2L}}$ be the set of all valid assignments on $\mathcal{T}_i^{2L}$, and let $\mathcal{C}_{\mathcal{T}_i^{2L}}(x) \triangleq \{\boldsymbol{w} \in \mathcal{C}_{\mathcal{T}_i^{2L}} : w_i = x, w_m = 0, \forall m \notin \mathcal{T}_i^{2L}\}$ be a subset of $\mathcal{C}_{\mathcal{T}_i^{2L}}$, where the assignment for every node $m \in \mathcal{V}_{\rm v} \setminus \mathcal{I}_{\rm v}$ is 0 and the assignment of the root node is x. In the remainder of this section, we often simplify $\mathcal{C}_{\mathcal{T}_i^{2L}}$ to $\mathcal{C}_{\mathcal{T}}$ when i and L are evident from the context. Similarly, we also simplify $\mathcal{C}_{\mathcal{T}_i^{2L}}(x)$ to $\mathcal{C}_{\mathcal{T}}(x)$.

3.2.2 Discounted Dynamic-Programming on a Tree

In this section, we recall the AttMP algorithm proposed in [58] for LDPC codes. When the number of iterations L satisfies $L < \frac{1}{4}\text{girth}(\mathcal{G})$, it can be considered as performing the AttMP algorithm on the computation trees, $\mathcal{T}_i^{2L}$, for all $i \in \mathcal{V}_{\rm v}$. As in [58], we consider the objective function in the logarithmic domain, and an alternative interpretation from the discounted DP point of view on the obtained message update rules is introduced.

Let $\gamma_m(x_m) \triangleq \log(p_{Y|X}(y_m|x_m))$ be the log-likelihood of receiving $y_m \in \mathbb{R}$ given

that $x_m \in \{0,1\}$ is the m-th transmitted codeword bit, and let $\beta \in \mathbb{R}_+$ be a non-negative weight factor. For a fixed $i \in \mathcal{V}_{\rm v}$, let $\mathcal{I}_{\rm v} \subseteq \mathcal{V}_{\rm v}$ be the subset of variable nodes associated with the variable nodes in a tree $\mathcal{T}_i^{2L}$. The AttMP algorithm solves the problem of finding a best assignment $\boldsymbol{w}^* \in \mathcal{C}_{\mathcal{T}}(0) \cup \mathcal{C}_{\mathcal{T}}(1)$ to $\mathcal{T}_i^{2L}$ so that the objective function

$$\sum_{m \in \mathcal{I}_{\rm v}} \beta_m \gamma_m(w_m), \tag{3.1}$$

is maximized, where $\beta_m = \beta^\ell$ if $m \in N(i, 2\ell)$. Since $\mathcal{C}_{\mathcal{T}}(0)$ and $\mathcal{C}_{\mathcal{T}}(1)$ are disjoint, (3.1) can be separated into two subproblems. For $x \in \{0,1\}$, define a function

$$\mu_i(x) \triangleq \max_{\boldsymbol{w} \in \mathcal{C}_{\mathcal{T}}(x)} \sum_{m \in \mathcal{I}_{\rm v}} \beta_m \gamma_m(w_m), \tag{3.2}$$

where $\mu_i(x)$ is the optimal reward for assigning x to the root node of $\mathcal{T}_i^{2L}$. Assume that the objective function (3.1) is uniquely maximized by $\boldsymbol{w}^*$. It can be shown that $w_i^* = \arg\max_{x \in \{0,1\}} \mu_i(x)$. Therefore, finding the best assignment of the tree $\mathcal{T}_i^{2L}$ is equivalent to finding the best assignment of the root node of $\mathcal{T}_i^{2L}$.

The RHS in (3.2) can be rewritten as

$$\mu_i(x) = \gamma_i(x) + \max_{\boldsymbol{w} \in \mathcal{C}_{\mathcal{T}}(x)} \sum_{\ell=1}^{L} \sum_{m \in N(i,2\ell)} \beta^\ell \gamma_m(w_m). \tag{3.3}$$

This suggests that we can compute $\mu_i(x)$ recursively by using DP. In the $(\ell+1)$-th iteration, we compute the optimal discounted total reward $\mu_{i \to j}^{(\ell+1)}(x)$ of assigning x

to the directed edge $i \to j$ by

$$\begin{aligned}\mu_{i\to j}^{(\ell+1)}(x) &\triangleq \gamma_i(x) + \beta \sum_{k\in N(i)\setminus j} \mu_{i\leftarrow k}^{(\ell)}(x) \\ &= \gamma_i(x) + \beta \sum_{k\in N(i)\setminus j} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \mu_{m\to k}^{(\ell)}(w_m), \end{aligned} \tag{3.4}$$

where

$$\mathcal{S}_{k,i}(x) \triangleq \Big\{\boldsymbol{w}\in\{0,1\}^n : w_i = x, \sum_{m\in N(k)} w_m = 0 \mod 2\Big\} \tag{3.5}$$

is the set of all valid assignments for the neighbors of the check node k when x is assigned to the i-th bit node. This follows from defining $\mu_{i\leftarrow k}^{(\ell)}(x)$ to be the optimal discounted total reward for assigning x to the directed edge $i \leftarrow k$ according to the rule

$$\mu_{i\leftarrow k}^{(\ell)}(x) \triangleq \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \mu_{m\to k}^{(\ell)}(w_m). \tag{3.6}$$

Finally, the reward function (3.3) can be computed by

$$\mu_i(x) = \gamma_i(x) + \beta \sum_{j\in N(i)} \mu_{i\leftarrow j}^{(L-1)}(x), \tag{3.7}$$

and the best assignment, or hard decision, to the root of the tree $\mathcal{T}_i^{2L}$ is then

$$\hat{x}_i \triangleq \arg\max_{x\in\{0,1\}} \mu_i(x). \tag{3.8}$$

To initialize the process, we choose $\mu_{i\to j}^{(0)}(x) = \gamma_i(x)$ for all edges $(i,j) \in \mathcal{E}$ and all $x \in \{0,1\}$. The update rule in (3.4) is the same as the AttMP algorithm proposed in [58], where the optimal discounted total rewards $\mu_{i\to j}^{(\ell)}(x)$ and $\mu_{i\leftarrow j}^{(\ell)}(x)$ are the

messages passed on the directed edges $i \to j$ and $i \leftarrow j$, respectively. Note that the obtained message update rule for the AttMP algorithm is actually an attenuated max-sum algorithm since the objective function as well as the message updates are considered in the logarithmic domain.

By using the message update rule (3.4), one can compute $\mu_i(x)$ for all $i \in \mathcal{V}_{\rm v}$ in parallel. Suppose that the total number of iterations L is less than $\frac{1}{4}\text{girth}(\mathcal{G})$. The vector $\hat{\boldsymbol{x}} = \{\hat{x}_i : i \in \mathcal{V}_{\rm v}\}$ is a collection of the best assignment to the root of the trees $\{\mathcal{T}_i^{2L} : i \in \mathcal{V}_{\rm v}\}$. In [68], the VERIFY-LO algorithm is proposed to test the local optimality of $\hat{\boldsymbol{x}}$. Note that we do not define local optimality in the dissertation. The interested reader can refer to [1] and [68] for the definition. After knowing that $\hat{\boldsymbol{x}}$ is locally optimal, it can be shown that $\hat{\boldsymbol{x}}$ is also an ML codeword [1, 67]. In this section, we discuss how the weight factor β affects the convergence of the AttMP decoding algorithm. When the AttMP decoding algorithm converges, a sufficient condition for testing the ML optimality of the corresponding hard-decision vector is proposed. It can be shown that checking the proposed sufficient condition is simpler than the VERIFY-LO algorithm. Finally, the analysis is extended to $L > \frac{1}{4}\text{girth}(\mathcal{G})$.

3.2.3 Attenuated Max-Product Decoding

In Section 3.2.2, the original AttMP algorithm was introduced. In this section, we introduce a modified version of the AttMP algorithm, which is mathematically equivalent to the original one for any finite number of iterations.

Let $\gamma_m \triangleq \gamma_m(0) - \gamma_m(1)$ be the channel log-likelihood ratio (LLR) for the m-th bit. Consider the computation tree $\mathcal{T}_i^{2L}$ of depth $2L$ with $L < \frac{1}{4}\text{girth}(\mathcal{G})$ and rooted at $i \in \mathcal{V}_{\rm v}$. Let $\mathcal{I}_{\rm v} \subseteq \mathcal{V}_{\rm v}$ be the subset of variable nodes associated with the variable nodes in a tree $\mathcal{T}_i^{2L}$. It can be shown that any assignment $\boldsymbol{w}^*(x)$ that maximizes the

objective function of (3.2) also maximizes the following objective function

$$\mu_i(x) \triangleq \max_{\boldsymbol{w}\in\mathcal{C}_{\mathcal{T}}(x)} \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m \left(1-w_m\right)\gamma_m, \tag{3.9}$$

where $\mu_i(x)$ is the optimal reward for assigning x to the root node of $\mathcal{T}_i^{2L}$. To show the equivalence between (3.2) and (3.9), we subtract a constant $\sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m\gamma_m(1)$ from the objective function of (3.2). Then,

$$\begin{aligned}
\arg\max_{\boldsymbol{w}\in\mathcal{C}_{\mathcal{T}}(x)} \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m\gamma_m(w_m) &= \arg\max_{\boldsymbol{w}\in\mathcal{C}_{\mathcal{T}}(x)} \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m\gamma_m(w_m) - \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m\gamma_m(1) \\
&= \arg\max_{\boldsymbol{w}\in\mathcal{C}_{\mathcal{T}}(x)} \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m \left(\gamma_m(w_m) - \gamma_m(1)\right) \\
&= \arg\max_{\boldsymbol{w}\in\mathcal{C}_{\mathcal{T}}(x)} \sum_{m\in\mathcal{I}_{\mathrm{v}}} \beta_m \left(1-w_m\right)\gamma_m.
\end{aligned}$$

Therefore, the modified AttMP update rule becomes

$$\begin{aligned}
\mu_{i\to j}^{(\ell+1)}(x) &\triangleq (1-x)\gamma_i + \beta \sum_{k\in N(i)\setminus j} \mu_{i\leftarrow k}^{(\ell)}(x) \\
&= (1-x)\gamma_i + \beta \sum_{k\in N(i)\setminus j} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \mu_{m\to k}^{(\ell)}(w_m),
\end{aligned} \tag{3.10}$$

and thus

$$\mu_{i\leftarrow k}^{(\ell)}(x) \triangleq \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \mu_{m\to k}^{(\ell)}(w_m). \tag{3.11}$$

After obtaining (3.11), we can compute the total reward function and the hard decision for the root node of the tree $\mathcal{T}_i^{2L}$ using (3.7) and (3.8), respectively. Note that the message $\mu_{i\to j}^{(\ell+1)}(x)$ now represents the weighted correlation between the LLRs and the best valid assignment with x assigned to the directed edge $i \to j$. The algorithm starts by setting $\mu_{i\to j}^{(0)}(x) = (1-x)\gamma_i$ for all $(i,j) \in \mathcal{E}$ and all $x \in \{0,1\}$.

Similar to the analysis in Section 3.2.2, $\mu_{i\to j}(x)$ can be considered as a DP value function, that assigns a real number to each bit-to-check directed edge $i \to j$ with all possible assignments $x \in \{0,1\}$. Since the messages can be updated in parallel, all bit-to-check messages with all possible assignments can be arranged as an AttMP message vector $\boldsymbol{\mu} \in \mathbb{R}^{2|\mathcal{E}|}$ defined by $\boldsymbol{\mu} \triangleq \{\mu_{i\to j}(x) : (i,j) \in \mathcal{E}, x \in \{0,1\}\}$. Based on the standard approach to DP [73, §6], the update process can be seen as applying an operator $\mathsf{A} : \mathbb{R}^{2|\mathcal{E}|} \to \mathbb{R}^{2|\mathcal{E}|}$ to message vectors. From (3.10), the operator A is defined by $\boldsymbol{\nu} = \mathsf{A}[\boldsymbol{\mu}]$ with

$$\begin{aligned}\nu_{i\to j}(x) &\triangleq (1-x)\gamma_i + \beta \sum_{k\in N(i)\backslash j} \mu_{i\leftarrow k}(x). \\ &= (1-x)\gamma_i + \beta \sum_{k\in N(i)\backslash j} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\backslash i} \mu_{m\to k}(w_m). \end{aligned} \tag{3.12}$$

The AttMP algorithm proceeds iteratively by computing $\boldsymbol{\mu}^{(\ell+1)} = \mathsf{A}[\boldsymbol{\mu}^{(\ell)}]$.

3.2.4 Weighted Min-Sum Decoding

Instead of passing the vector $(\mu^{(\ell)}_{i\to j}(0), \mu^{(\ell)}_{i\to j}(1)) \in \mathbb{R}^2$ as the $i \to j$ message in the AttMP algorithm, the WMS algorithm passes the message $\mu^{(\ell)}_{i\to j} \triangleq \mu^{(\ell)}_{i\to j}(0) - \mu^{(\ell)}_{i\to j}(1)$, which is simply the difference between the best 0-root correlation and the best 1-root correlation. Similarly, the $i \leftarrow j$ message is simplified to $\mu^{(\ell)}_{i\leftarrow j} \triangleq \mu^{(\ell)}_{i\leftarrow j}(0) - \mu^{(\ell)}_{i\leftarrow j}(1)$. The message update rules of the WMS algorithm are therefore given by

$$\mu^{(\ell+1)}_{i\to j} \triangleq \gamma_i + \beta \sum_{k\in N(i)\backslash j} \mu^{(\ell)}_{i\leftarrow k}, \tag{3.13}$$

$$\mu^{(\ell)}_{i\leftarrow j} \triangleq \left(\prod_{m\in N(j)\backslash i} \operatorname{sgn}\left(\mu^{(\ell)}_{m\to j}\right) \right) \min_{m'\in N(j)\backslash i} \left|\mu^{(\ell)}_{m'\to j}\right|, \tag{3.14}$$

where $\mathrm{sgn}(x)$ is the standard sign function defined by

$$\mathrm{sgn}(x) \triangleq \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases}$$

Define $\mu_i^{(\ell)} \triangleq \gamma_i + \beta \sum_{j \in N(i)} \mu_{i \leftarrow j}^{(\ell)}$. The hard decision of the i-th bit node after ℓ iterations is

$$\hat{x}_i = \begin{cases} \frac{1}{2}\left(1 - \mathrm{sgn}\left(\mu_i^{(\ell)}\right)\right) & \text{if } \mu_i^{(\ell)} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

Note that $\mu_i^{(\ell)} = 0$ implies $\mu_i^{(\ell)}(0) = \mu_i^{(\ell)}(1)$. So, the decoder can either assign 0 or 1 to the i-th bit. In the practical implementation of the WMS algorithm, ties can be broken by randomly assigning 0 or 1 with probability $\frac{1}{2}$. In the present analysis, we avoid this case by assuming that $\mathrm{sgn}\left(\mu_i^{(\ell)}\right) \neq 0$ when computing hard decisions.

Similar to the AttMP algorithm, we define the WMS message vector $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{E}|}$ by $\boldsymbol{\mu} \triangleq \{\mu_{i \to j} : (i,j) \in \mathcal{E}\}$. The message update rule of the WMS algorithm is also given by the operator $\mathsf{W} : \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}^{|\mathcal{E}|}$, which is defined by $\boldsymbol{\nu} = \mathsf{W}[\boldsymbol{\mu}]$ with

$$\nu_{i \to j} = \gamma_i + \beta \sum_{k \in N(i) \setminus j} \left(\prod_{m \in N(k) \setminus i} \mathrm{sgn}\left(\mu_{m \to k}\right) \right) \min_{m' \in N(k) \setminus i} \left|\mu_{m' \to k}\right|. \tag{3.16}$$

The WMS algorithm is initialized by setting $\mu_{i \to j}^{(0)} = \gamma_i$ and proceeds iteratively by computing $\boldsymbol{\mu}^{(\ell+1)} = \mathtt{W}[\boldsymbol{\mu}^{(\ell)}]$.

3.2.5 LP Decoding

Given the received vector $\boldsymbol{y} \in \mathbb{R}^n$, the ML decoder finds a codeword $\boldsymbol{x}^* \in \mathcal{C}$ such that the probability $p(\boldsymbol{y}|\boldsymbol{x}^*)$ is maximal among all $\boldsymbol{x} \in \mathcal{C}$. Let $\boldsymbol{\gamma} \in \mathbb{R}^n$ be the vector of channel LLRs. Then, ML decoding can be rewritten as the following integer programming problem [60],

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} x_i \gamma_i \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{C}. \end{aligned} \tag{3.17}$$

Note that the problem (3.17) is equivalent to the problem of $\max \sum_{i=1}^{n} -x_i \gamma_i$ subject to $\boldsymbol{x} \in \mathcal{C}$. By the fact that adding the constant, $\sum_{i=1}^{n} \gamma_i$, to the objective function does not change the solution, the problem (3.17) is then equivalent to

$$\begin{aligned} \max \quad & \textstyle\sum_{i=1}^{n} (1 - x_i) \gamma_i \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{C}. \end{aligned} \tag{3.18}$$

For LDPC codes, solving (3.17) or (3.18) directly is computationally infeasible for large n because the number of codewords grows exponentially in n. In [60], a sub-optimal decoder, the so called LP decoder, was proposed. With the same objective function as in (3.17), the LP decoder searches the optimal solution over a relaxed polytope which is obtained by intersecting all local codeword polytopes defined by each check node of the graph $\mathcal{G}$.

Here, we briefly describe the LP decoder in [60] as follows. Given a check node $j \in \mathcal{V}_{\mathrm{c}}$, let

$$\mathcal{E}_j = \{S \subseteq N(j) : |S| \text{ is even}\}$$

be the collection of all support sets of local codewords for j. Note that $\emptyset \in \mathcal{E}_j$ and

represents the all-zero codeword. For each $j \in \mathcal{V}_{\rm c}$, and $S \in \mathcal{E}_j$, $\zeta_{j,S}$ is an indicator function of the local codeword being assigned to j. The LP decoder solves the following problem

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{V}_{\rm v}} x_i \gamma_i \\
\text{subject to} \quad & \sum_{S \in \mathcal{E}_j} \zeta_{j,S} = 1 \quad \forall j \in \mathcal{V}_{\rm c} \\
& \sum_{\substack{S \in \mathcal{E}_j \\ S \ni i}} \zeta_{j,S} = x_i \quad \forall (i,j) \in \mathcal{E} \\
& \zeta_{j,S} \geq 0, \qquad \forall j \in \mathcal{V}_{\rm c},\ \forall S \in \mathcal{E}_j.
\end{aligned}$$

In the sequel, this LP problem is called *Problem-P*. The vector $\boldsymbol{x}^*$ is *LP optimal* if it is the solution of *Problem-P*. Moreover, if $\boldsymbol{x}^* \in \{0,1\}^n$, then the vector $\boldsymbol{x}^*$ is an ML codeword.

To establish the dual problem of *Problem-P*, a Lagrange multiplier $\tau_{i,j}$ is associated with each edge $(i,j) \in \mathcal{E}$ of the graph $\mathcal{G}$. The resulting dual problem is given by

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{V}_{\rm c}} \tau_j \\
\text{subject to} \quad & \sum_{i \in S} \tau_{i,j} \geq \tau_j \quad \forall j \in \mathcal{V}_{\rm c},\ \forall S \in \mathcal{E}_j \\
& \sum_{j \in N(i)} \tau_{i,j} \leq \gamma_i \qquad \forall i \in \mathcal{V}_{\rm v},
\end{aligned}$$

which, as shown in [66], is equivalent to

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{V}_{\rm c}} \min_{S \in \mathcal{E}_j} \sum_{i \in S} \tau_{i,j} \\
\text{subject to} \quad & \sum_{j \in N(i)} \tau_{i,j} = \gamma_i \quad \forall i \in \mathcal{V}_{\rm v}.
\end{aligned}$$

In the remainder of thissection, this dual problem is called *Problem-D*.

Consider a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code, and let

$$\mathcal{L} \triangleq \left\{ \boldsymbol{w} \in \{0,1\}^{d_{\mathrm{c}}} : \sum_{i=1}^{d_{\mathrm{c}}} w_i = 0 \mod 2 \right\} \tag{3.19}$$

be the set of all valid codeword of a check node. For each check node $j \in \mathcal{V}_{\mathrm{c}}$, define a mapping $\varphi_j : \{1, 2, \ldots, d_{\mathrm{c}}\} \to N(j)$ such that $\varphi_j(1) < \varphi_j(2) < \cdots < \varphi_j(d_{\mathrm{c}})$. Let $\boldsymbol{\tau}_j \in \mathbb{R}^{d_{\mathrm{c}}}$ be a vector with $(\boldsymbol{\tau}_j)_m = \tau_{\varphi_j(m),j}$ for $m = 1, 2, \ldots, d_{\mathrm{c}}$. The objective function in *Problem-D* can be rewritten as $\sum_{j \in \mathcal{V}_{\mathrm{c}}} \min_{\boldsymbol{w} \in \mathcal{L}} \langle \boldsymbol{w}, \boldsymbol{\tau}_j \rangle$, where $\langle \boldsymbol{w}, \boldsymbol{\tau}_j \rangle \triangleq \sum_{m=1}^{d_{\mathrm{c}}} w_m (\boldsymbol{\tau}_j)_m$.

3.2.6 Impossibility of a General ML Certificate for WMS Decoding

One of the main goals of this section, is to show that the WMS algorithm with $\beta < \frac{1}{d_{\mathrm{v}}-1}$ converges to a fixed point, and returns an ML codeword if the fixed point satisfies the proposed consistency conditions. Before we prove this, two examples are provided in this section for showing that the WMS algorithm with some $\beta > \frac{1}{d_{\mathrm{v}}-1}$ is not guaranteed to return an ML codeword.

The first example is a numerical simulation of the WMS algorithm with $\beta = 0.8$ on a small code ($n = 12$). Since the codeword length is short, we are able to implement the ML decoder defined in (3.17) and compare the ML output with the WMS output.

Example 34. In this example, the ML optimality of the codeword returned by the WMS decoder with $\beta = 0.8$ is checked. We consider a $(3, 4)$-regular LDPC code over the BSC with cross-over probability $p = 0.1$. The parity check matrix for the

$(3, 4)$-regular LDPC code is

$$H = \begin{pmatrix} 0,\,0,\,0,\,1,\,1,\,0,\,1,\,0,\,0,\,0,\,0,\,1 \\ 0,\,0,\,0,\,0,\,0,\,1,\,1,\,1,\,0,\,1,\,0,\,0 \\ 0,\,1,\,0,\,1,\,0,\,1,\,0,\,0,\,1,\,0,\,0,\,0 \\ 1,\,1,\,1,\,0,\,0,\,0,\,1,\,0,\,0,\,0,\,0,\,0 \\ 1,\,0,\,0,\,1,\,0,\,0,\,0,\,0,\,0,\,1,\,1,\,0 \\ 0,\,0,\,1,\,0,\,1,\,1,\,0,\,0,\,0,\,0,\,1,\,0 \\ 1,\,0,\,0,\,0,\,1,\,0,\,0,\,1,\,1,\,0,\,0,\,0 \\ 0,\,1,\,0,\,0,\,0,\,0,\,0,\,1,\,0,\,0,\,1,\,1 \\ 0,\,0,\,1,\,0,\,0,\,0,\,0,\,0,\,1,\,1,\,0,\,1 \end{pmatrix}.$$

For the WMS decoder, 200 iterations were performed in decoding each block. After testing 10^5 blocks, there were 90905 codewords returned by the WMS decoder. Among these codewords returned by the WMS algorithm, only 90850 codewords were the ML codeword. Therefore, codewords returned by the WMS algorithm with $\beta = 0.8$ cannot be guaranteed to be ML optimal.

For the general case, the following example gives some intuition.

Example 35. Consider a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code with codeword length n, where d_{c} is an odd number and $d_{\mathrm{v}} > 3$. Assume that the all-zero codeword is transmitted. Let the channel output LLR vector be $\boldsymbol{\gamma} = (-1, \ldots, -1)$. Consider the WMS algorithm with $\beta > \frac{2}{d_{\mathrm{v}}-1}$.

At the beginning, all LLR-based messages from variable nodes to their neighboring check nodes are $\mu_{i\to j}^{(0)} = -1$ for $i = 1, \ldots, n$ and $j \in N(i)$. Consider messages $\mu_{i\leftarrow j}$ passed from the j-th check node to its neighbor variable nodes $i \in N(j)$. Since the incoming messages are all equal to -1, the message update rule of the WMS

algorithm at the check node gives

$$\mu_{i\leftarrow j}^{(0)} = \left(\prod_{k\in N(j)\setminus i} \operatorname{sgn}(\mu_{k\to j}) \right) \min_{k'\in N(j)\setminus i} \left| \mu_{k'\to j}^{(0)} \right| = 1,$$

for all $(i,j) \in \mathcal{E}$. In the first iteration, the outgoing message from the i-th variable node to the j-th check node is therefore

$$\mu_{i\to j}^{(1)} = \gamma_i + \beta \sum_{k\in N(i)\setminus j} \mu_{i\leftarrow k}^{(0)} > -1 + (d_{\mathrm{v}} - 1)\frac{2}{d_{\mathrm{v}} - 1} = 1.$$

Moreover, one can show that $\mu_{i\to j}^{(\ell)} \to \infty$ as $\ell \to \infty$. Thus, the hard decisions return the all-zero codeword. Unfortunately, given this $\boldsymbol{\gamma}$, we know that the ML output must be a nonzero codeword with maximal Hamming weight. Therefore, the WMS algorithm cannot provide an ML certificate for $\beta > \frac{2}{d_{\mathrm{v}}-1}$. One might worry that this effect may be related to ties between ML codewords, but these can be avoided, without affecting the above result, by adding a very small amount of uniform random noise to the channel output LLR vector.

3.3 Convergence and Optimality Guarantees

For the AttMP algorithm on $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes with the weight factor β satisfying $\beta(d_{\mathrm{v}} - 1)(d_{\mathrm{c}} - 1) < 1$, Frey and Koetter [58] showed that it will return the ML codeword if it converges to a codeword. However, the convergence of the AttMP algorithm, when $\beta(d_{\mathrm{v}} - 1)(d_{\mathrm{c}} - 1) < 1$, is not shown in their work. In this section, we first show that the AttMP algorithm will converge to a fixed point by using the contraction mapping theorem [72, pp. 97–100]. Then, we introduce another proof showing that the codeword obtained at the fixed point is the ML codeword if the fixed point satisfies the proposed consistency conditions. We also analyze the convergence of the WMS algorithm. Compared to the convergence analysis of the

AttMP algorithm, a weaker condition for the convergence of the WMS algorithm, namely $\beta(d_{\mathrm{v}}-1)<1$, is obtained. We also show that, if the converged messages satisfy the consistency conditions, which are similar to the conditions for the AttMP algorithm, the LP, and therefore ML, optimality of the WMS codeword is guaranteed.

3.3.1 Attenuated Max-product Decoding

Before showing that the AttMP algorithm converges to a fixed point when $\beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$, we first introduce the following auxiliary lemma.

Lemma 36. *For any two vectors $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^n$, the following inequality holds*

$$\max_i |f_i - g_i| \geq \left| \max_i f_i - \max_{i'} g_{i'} \right|. \tag{3.20}$$

Proof. Let $1 \leq \ell, m \leq n$ be the integers such that $f_\ell = \max_i f_i$ and $g_m = \max_i g_i$. If $f_\ell \geq g_m$, we know that $f_\ell - g_i \geq 0$ for all $i = 1, 2, \ldots, n$. It follows that

$$\max_i |f_i - g_i| \geq |f_\ell - g_\ell| = f_\ell - g_\ell \geq f_\ell - g_m = |f_\ell - g_m|.$$

On the other hand, if $f_\ell \leq g_m$, a symmetric argument shows $\max_i |f_i - g_i| \geq |f_\ell - g_m|$ as well. Therefore, we obtain (3.20).

Theorem 37. *For any $0 \leq \beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$, the operator $\mathtt{A}$ is a contraction on $\left(\mathbb{R}^{2|\mathcal{E}|}, \|\cdot\|_\infty\right)$. Therefore, the AttMP algorithm with $0 \leq \beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$ converges to a unique fixed point as $\ell \to \infty$.*

Proof. Let $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{2|\mathcal{E}|}$ be two AttMP message vectors, and let $\boldsymbol{\mu}' = \mathtt{A}[\boldsymbol{\mu}]$ and $\boldsymbol{\nu}' = \mathtt{A}[\boldsymbol{\nu}]$. By the definition of $\mathtt{A}$ in (3.12) and the triangle inequality, $\|\boldsymbol{\mu}' - \boldsymbol{\nu}'\|_\infty$can

be upper bounded by

$$\begin{aligned}\left\|\boldsymbol{\mu}'-\boldsymbol{\nu}'\right\|_{\infty} &= \beta \max_{x\in\{0,1\},\,(i,j)\in\mathcal{E}} \left| \sum_{k\in N(i)\setminus j} \mu_{i\leftarrow k}(x) - \sum_{k'\in N(i)\setminus j} \nu_{i\leftarrow k'}(x) \right| \\ &\leq \beta \max_{x\in\{0,1\},\,(i,j)\in\mathcal{E}} \sum_{k\in N(i)\setminus j} \left|\mu_{i\leftarrow k}(x)-\nu_{i\leftarrow k}(x)\right|. \end{aligned} \tag{3.21}$$

From (3.6), the last term of the RHS in (3.21) can be rewritten as

$$\begin{aligned}\left|\mu_{i\leftarrow k}(x)-\nu_{i\leftarrow k}(x)\right| &= \left| \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \mu_{m\to k}(w_m) - \max_{\boldsymbol{w}'\in\mathcal{S}_{k,i}(x)} \sum_{m'\in N(k)\setminus i} \nu_{m'\to k}(w'_{m'}) \right| \\ &\overset{\text{(a)}}{\leq} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \left| \sum_{m\in N(k)\setminus i} \mu_{m\to k}(w_m) - \sum_{m\in N(k)\setminus i} \nu_{m\to k}(w_m) \right| \\ &\leq \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \left|\mu_{m\to k}(w_m)-\nu_{m\to k}(w_m)\right|, \end{aligned}$$

where the inequality (a) follows by Lemma 36. Thus, the RHS in (3.21) is upper bounded by

$$\beta \max_{x\in\{0,1\},\,(i,j)\in\mathcal{E}} \sum_{k\in N(i)\setminus j} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \sum_{m\in N(k)\setminus i} \left|\mu_{m\to k}\left(w_m\right)-\nu_{m\to k}\left(w_m\right)\right|. \tag{3.22}$$

Since $\max(\boldsymbol{f}+\boldsymbol{g}) \leq \max \boldsymbol{g} + \max \boldsymbol{f}$, Equation (3.22) is less than or equal to

$$\beta \max_{x\in\{0,1\},\,(i,j)\in\mathcal{E}} \sum_{k\in N(i)\setminus j} \sum_{m\in N(k)\setminus i} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \left|\mu_{m\to k}\left(w_m\right)-\nu_{m\to k}\left(w_m\right)\right|. \tag{3.23}$$

By the fact that, for any $m \in N(k)\setminus i$,

$$\max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x)} \left|\mu_{m\to k}\left(w_m\right)-\nu_{m\to k}\left(w_m\right)\right| = \max_{z\in\{0,1\}} \left|\mu_{m\to k}\left(z\right)-\nu_{m\to k}\left(z\right)\right|$$

and

$$\begin{aligned}\max_{z\in\{0,1\}} \left|\mu_{m\to k}(z) - \nu_{m\to k}(z)\right| &\le \max_{(i',j')\in\mathcal{E}} \max_{x'\in\{0,1\}} \left|\mu_{i'\to j'}(x') - \nu_{i'\to j'}(x')\right| \\ &= \left\|\boldsymbol{\mu} - \boldsymbol{\nu}\right\|_\infty,\end{aligned}$$

Equation (3.23) is further upper bounded by

$$\begin{aligned}&\beta \max_{x\in\{0,1\},\,(i,j)\in\mathcal{E}} \sum_{k\in N(i)\setminus j} \sum_{m\in N(k)\setminus i} \left\|\boldsymbol{\mu} - \boldsymbol{\nu}\right\|_\infty \\ &\overset{\text{(a)}}{=} \beta\,(d_\text{v}-1)\,(d_\text{c}-1)\left\|\boldsymbol{\mu} - \boldsymbol{\nu}\right\|_\infty \\ &\overset{\text{(b)}}{<} \left\|\boldsymbol{\mu} - \boldsymbol{\nu}\right\|_\infty,\end{aligned}$$

where the equality (a) holds since $\left\|\boldsymbol{\mu} - \boldsymbol{\nu}\right\|_\infty$ is independent of (x, i, j, m, k), and the inequality (b) follows from the fact that $\beta(d_\text{c}-1)(d_\text{v}-1) < 1$. This shows the contraction.

The second part of the theorem follows from the contraction mapping theorem.

Remark 38. Theorem 37 shows that, for an arbitrary (d_v, d_c)-regular LDPC code and any $0 \le \beta < \frac{1}{(d_\text{c}-1)(d_\text{v}-1)}$, the AttMP algorithm converges to a unique fixed point denoted by $\boldsymbol{\mu}^*$. That is, $\boldsymbol{\mu}^{(\ell)} \to \boldsymbol{\mu}^*$ as $\ell \to \infty$, and $\boldsymbol{\mu}^* = \mathsf{A}[\boldsymbol{\mu}^*]$. In the context of the DP, the contraction mapping theorem is also employed to show the existence of optimal stationary policies for discounted Markov decision processes [73, §6.2].

For each $(i, j) \in \mathcal{E}$, let $x^*_{i,j} \in \{0, 1\}$ be the assignment that uniquely maximizes the converged message $\mu^*_{i\to j}(x)$, and let $\boldsymbol{x}^* \in \{0, 1\}^n$ be the vector returned by the AttMP algorithm. In [58], Frey and Koetter showed that $\boldsymbol{x}^*$ is the ML codeword if $x^*_{i,j} = x^*_i$ for all $i \in \mathcal{V}_\text{v}$ and $j \in N(i)$, and $\boldsymbol{x}^* \in \mathcal{C}$. We first state these conditions formally in Definition 39. Then, using the equivalent objective function in (3.9), an

easier proof of showing the ML optimality of $\boldsymbol{x}^*$ is introduced.

Definition 39 (AttMP-consistency)**.** The assignment $\{x^*_{i,j} : (i,j) \in \mathcal{E}\}$ is called *AttMP-consistent* if $x^*_{i,j} = x^*_i$ for all $i \in \mathcal{V}_{\rm v}$ and $j \in N(i)$ and if the vector $\boldsymbol{x}^*$ with components x^*_i is a codeword.

Lemma 40. *Consider a $(d_{\rm v}, d_{\rm c})$-regular LDPC code, and choose $\beta < \frac{1}{(d_{\rm v}-1)(d_{\rm c}-1)}$. Let $\boldsymbol{\mu}^*$ be a fixed point of the AttMP algorithm, and let $x^*_{i,j}$ uniquely maximize $\mu^*_{i\to j}(x)$ for each edge $(i,j) \in \mathcal{E}$. Then for any binary vector $\{x_{i,j}\} \in \{0,1\}^{|\mathcal{E}|}$,*

$$\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x_{i,j}\right) \le \sum_{(i,j)\in\mathcal{E}} \left(1 - x_{i,j}\right)\gamma_i + \beta\left(d_{\rm v}-1\right)\left(d_{\rm c}-1\right) \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right), \quad (3.24)$$

with equality if and only if $\{x^*_{i,j} : (i,j) \in \mathcal{E}\}$ is AttMP-consistent.

Proof. By the definition of the DP value function in (3.10), we have

$$\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x_{i,j}\right) = \sum_{(i,j)\in\mathcal{E}} \left(1 - x_{i,j}\right)\gamma_i + \beta \sum_{\substack{(i,j)\in\mathcal{E}\\ k\in N(i)\setminus j}} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x_{i,j})} \sum_{m\in N(k)\setminus i} \mu^*_{m\to k}\left(w_m\right), \quad (3.25)$$

where $\mathcal{S}_{k,i}(x_{i,j})$ is defined in (3.5). Since $x^*_{i,j}$ maximizes $\mu^*_{i\to j}(x)$ for all $(i,j) \in \mathcal{E}$, by replacing w_m in (3.25) with $x^*_{m,k}$, we obtain

$$\begin{aligned}\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}(x_{i,j}) &\le \sum_{(i,j)\in\mathcal{E}} (1 - x_{i,j})\gamma_i + \beta \sum_{\substack{(i,j)\in\mathcal{E},\, k\in N(i)\setminus j,\\ m\in N(k)\setminus i}} \mu^*_{m\to k}\left(x^*_{m,k}\right)\\ &= \sum_{(i,j)\in\mathcal{E}} \left(1 - x_{i,j}\right)\gamma_i + \beta\left(d_{\rm v}-1\right)\left(d_{\rm c}-1\right) \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right),\end{aligned}$$

which is the promised inequality in (3.24). To show the equality in (3.24), by sub-

stituting $x^*_{i,j}$ into (3.25), we have

$$\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right) = \sum_{(i,j)\in\mathcal{E}} \left(1-x^*_{i,j}\right)\gamma_i + \beta \sum_{\substack{(i,j)\in\mathcal{E}\\ k\in N(i)\backslash j}} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(x^*_{i,j})} \sum_{m\in N(k)\backslash i} \mu^*_{m\to k}\left(w_m\right). \tag{3.26}$$

Since $\{x^*_{i,j}\}$ is AttMP-consistent, there exists a vector $\boldsymbol{x}^* \in \mathcal{C}$ such that $x^*_i = x^*_{i,j}$ for all $i \in \mathcal{V}_{\rm v}$ and $j \in N(i)$. By the fact that $\boldsymbol{x}^* \in \mathcal{S}_{k,i}(x^*_{i,j})$, the last term in (3.26) is equal to

$$\beta \sum_{\substack{(i,j)\in\mathcal{E},\, k\in N(i)\backslash j,\\ m\in N(k)\backslash i}} \mu^*_{m\to k}\left(x^*_{m,k}\right).$$

Therefore, we obtain the equality.

Remark 41. From Lemma 40, we know that, when the assignment $\{x^*_{i,j}\}$ is AttMP-consistent, then

$$\sum_{(i,\,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right) = \frac{d_{\rm v}\sum_{i\in\mathcal{V}_{\rm v}}\left(1-x^*_i\right)\gamma_i}{1-\beta\left(d_{\rm v}-1\right)\left(d_{\rm c}-1\right)}, \tag{3.27}$$

where $x^*_i = x^*_{i,j}$ for all $i \in \mathcal{V}_{\rm v}$ and $(i,j) \in \mathcal{E}$.

Theorem 42. *Given the LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$ and $\beta < \frac{1}{(d_{\rm v}-1)(d_{\rm c}-1)}$, let the assignment $x^*_{i,j}$ uniquely maximize $\mu^*_{i\to j}(x)$ for all $(i,j) \in \mathcal{E}$. If $\{x^*_{i,j} : (i,j) \in \mathcal{E}\}$ is AttMP-consistent, let $\boldsymbol{x}^* \triangleq \{x^*_i : i \in \mathcal{V}_{\rm v}\}$ with $x^*_i = x^*_{i,j}$ for any $j \in N(i)$. Then, $\boldsymbol{x}^*$ is the ML codeword.*

Proof. We prove that $\boldsymbol{x}^*$ is the ML codeword by showing that $\boldsymbol{x}^*$ uniquely maximizes the correlation $\sum_{i\in\mathcal{V}_{\rm v}}(1-x^*_i)\gamma_i$ over all codewords in $\mathcal{C}$.

Consider any codeword $\tilde{\boldsymbol{x}} \in \mathcal{C}$ such that $\tilde{\boldsymbol{x}} \neq \boldsymbol{x}^*$, and let $\{\tilde{x}_{i,j}\} \in \{0,1\}^{|\mathcal{E}|}$ be the corresponding binary vector with $\tilde{x}_{i,j} = \tilde{x}_i$ for all $j \in N(i)$. From (3.10), we know

that

$$\sum_{(i,j)\in\mathcal{E}} (1-\tilde{x}_{i,j})\,\gamma_i = \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\,(\tilde{x}_{i,j}) - \beta \sum_{\substack{(i,j)\in\mathcal{E}\\ k\in N(i)\backslash j}} \max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(\tilde{x}_{i,j})} \sum_{m\in N(k)\backslash i} \mu^*_{m\to k}\,(w_m)\,. \tag{3.28}$$

By the fact that $\tilde{\boldsymbol{x}}$ is also in $\mathcal{S}_{k,i}(\tilde{x}_{i,j})$, we have

$$\max_{\boldsymbol{w}\in\mathcal{S}_{k,i}(\tilde{x}_{i,j})} \sum_{m\in N(k)\backslash i} \mu^*_{m\to k}\,(w_m) \geq \sum_{m\in N(k)\backslash i} \mu^*_{m\to k}\,(\tilde{x}_{m,k})\,.$$

Therefore, the RHS in (3.28) is upper bounded by

$$\begin{aligned}&\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\,(\tilde{x}_{i,j}) - \beta \sum_{\substack{(i,j)\in\mathcal{E},\, k\in N(i)\backslash j\\ m\in N(k)\backslash i}} \mu^*_{m\to k}\,(\tilde{x}_{m,k})\\ &\qquad = (1-\beta\,(d_{\mathrm{v}}-1)\,(d_{\mathrm{c}}-1)) \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\,(\tilde{x}_{i,j})\,.\end{aligned} \tag{3.29}$$

Since $x^*_{i,j}$ uniquely maximizes $\mu^*_{i\to j}\,(x)$ for all $(i,j)\in\mathcal{E}$, the vector $\{x^*_{i,j} : (i,j)\in\mathcal{E}\}$ also uniquely maximizes $\sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\,(x)$. The RHS in (3.29) is less than

$$(1-\beta\,(d_{\mathrm{v}}-1)\,(d_{\mathrm{c}}-1)) \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right).$$

Thus, we have

$$\begin{aligned}\sum_{i\in\mathcal{V}_{\mathrm{v}}} (1-\tilde{x}_i)\,\gamma_i &< \frac{1}{d_{\mathrm{v}}}\,(1-\beta\,(d_{\mathrm{v}}-1)\,(d_{\mathrm{c}}-1)) \sum_{(i,j)\in\mathcal{E}} \mu^*_{i\to j}\left(x^*_{i,j}\right)\\ &\overset{\text{(a)}}{=} \sum_{i\in\mathcal{V}_{\mathrm{v}}} (1-x^*_i)\,\gamma_i,\end{aligned}$$

where (a) follows from (3.27). This shows that $\boldsymbol{x}^*$ uniquely maximizes the correlation

$\sum_{i\in\mathcal{V}_{\rm v}}(1-x_i)\,\gamma_i$ over all $\boldsymbol{x}\in\mathcal{C}$. From (3.18), we know that $\boldsymbol{x}^*$ is the ML codeword.

3.3.2 Weighted Min-Sum Decoding

We first introduce the consistency condition for the WMS decoding algorithm. Then, we prove this consistency condition is a sufficient condition for showing the ML optimality of the WMS codeword. In the following discussion, we use the vector $\boldsymbol{\mu}^{(\ell)}$ with the variable-to-check node messages at iteration ℓ to implicitly also define the vector with the check-to-variable node messages at iteration ℓ.

Definition 43 (WMS-consistency)**.** Let $\mu_{i\to j}^{(\ell)}$, defined in (3.13), be the message passed from the i-th bit node to the j-th check node in the first half of the ℓ-th iteration, and $\mu_{i\leftarrow j}^{(\ell)}$, defined in (3.14), be the message passed from the j-th check node to the i-th bit node in the second half of the ℓ-th iteration. The message vector $\boldsymbol{\mu}^{(\ell)}$ is called *WMS-consistent* if, for each bit $i\in\mathcal{V}_{\rm v}$, it satisfies

$$\gamma_i+\beta\sum_{j\in N(i)}\mu_{i\leftarrow j}^{(\ell)}\neq 0,$$

and

$$\operatorname{sgn}\left(\mu_{i\to j}^{(\ell)}\right)=\operatorname{sgn}\left(\mu_{i\leftarrow j}^{(\ell)}\right)=\operatorname{sgn}\left(\gamma_i+\beta\sum_{j'\in N(i)}\mu_{i\leftarrow j'}^{(\ell)}\right) \tag{3.30}$$

for all $j\in N(i)$.

When the WMS message vector is WMS-consistent, the following theorem shows that the corresponding hard decisions define a codeword.

Theorem 44. *If the WMS message vector in the ℓ-th iteration is WMS-consistent, then the hard decision vector $\hat{\boldsymbol{x}}\triangleq\{\hat{x}_i : i\in\mathcal{V}_{\rm v}\}$ with $\hat{x}_i$ defined in (3.15) gives a codeword.*

Proof. We prove this result by a contradiction. Suppose that $\hat{\boldsymbol{x}}$ is not a codeword. There exists at least one unsatisfied check node. Let $j \in \mathcal{V}_{\mathrm{c}}$ be an unsatisfied check node. Since $\sum_{i \in N(j)} \hat{x}_i = 1 \mod 2$, we have

$$\begin{aligned} -1 &= \prod_{i \in N(j)} \operatorname{sgn}\left(\gamma_i + \beta \sum_{j' \in N(i)} \mu_{i \leftarrow j'}^{(\ell)}\right) \\ &\overset{\text{(a)}}{=} \prod_{i \in N(j)} \operatorname{sgn}\left(\mu_{i \to j}^{(\ell)}\right), \end{aligned}$$

where (a) follows from (3.30). Consider the message passed from the j-th check to the i-th bit. From the WMS message update rule (3.14),

$$\begin{aligned} \mu_{i \leftarrow j}^{(\ell)} &= \left(\prod_{m \in N(j) \setminus i} \operatorname{sgn}\left(\mu_{m \to j}^{(\ell)}\right)\right) \min_{m \in N(j) \setminus i} \left|\mu_{m \to j}^{(\ell)}\right| \\ &= -\operatorname{sgn}\left(\mu_{i \to j}^{(\ell)}\right) \min_{m \in N(j) \setminus i} \left|\mu_{m \to j}^{(\ell)}\right|. \end{aligned}$$

This contradicts the first equality in (3.30). Therefore, we conclude that $\hat{\boldsymbol{x}}$ is a codeword.

Next, we consider the ML optimality of the solution returned by the WMS algorithm. Similar to the analysis of the AttMP algorithm, we first discuss the convergence of the WMS messages. When the WMS messages converge to a fixed point, we show that the implied hard decisions give an optimal codeword if the fixed point is WMS-consistent.

To show the convergence of the WMS algorithm, we first introduce the following lemma.

Lemma 45. *Consider two WMS message vectors* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{E}|}$. *Let* $i \in \mathcal{V}_{\mathrm{v}}$, $k \in \mathcal{V}_{\mathrm{c}}$

with $(i,k) \in \mathcal{E}$*, and define*

$$D_{i,k} \triangleq \left| \left(\prod_{m \in N(k) \backslash i} \operatorname{sgn}(\mu_{m \to k}) \right) \min_{m' \in N(k) \backslash i} |\mu_{m' \to k}| - \left(\prod_{m \in N(k) \backslash i} \operatorname{sgn}(\nu_{m \to k}) \right) \min_{m' \in N(k) \backslash i} |\nu_{m' \to k}| \right|.$$

Then,

$$\max_{m \in N(k) \backslash i} |\mu_{m \to k} - \nu_{m \to k}| \geq D_{i,k}.$$

Proof. See Appendix A.7.

To show the convergence of the WMS messages, it will suffice to show that the WMS operator $\mathtt{W}$ is an $\|\cdot\|_\infty$ contraction. The following theorem provides a precise statement.

Theorem 46. *For all LLR vectors and message vectors, the WMS operator* $\mathtt{W}$ *is a contraction on* $\left(\mathbb{R}^{|\mathcal{E}|}, \|\cdot\|_\infty\right)$ *if* $0 \leq \beta < \frac{1}{d_{\mathrm{v}}-1}$*. Therefore, the WMS algorithm with* $0 \leq \beta < \frac{1}{d_{\mathrm{v}}-1}$ *converges to a unique fixed point as* $\ell \to \infty$*.*

Proof. Let $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{E}|}$ be two WMS message vectors. Using Lemma 45, one can

upper bound $\|\mathtt{W}[\boldsymbol{\mu}] - \mathtt{W}[\boldsymbol{\nu}]\|_\infty$ as follows

$$\begin{aligned}
\|\mathtt{W}[\boldsymbol{\mu}] - \mathtt{W}[\boldsymbol{\nu}]\|_\infty = \max_{(i,j)\in\mathcal{E}} \Bigg| & \beta \sum_{k\in N(i)\setminus j} \left(\prod_{m\in N(k)\setminus i} \operatorname{sgn}(\mu_{m\to k}) \right) \min_{m'\in N(k)\setminus i} |\mu_{m'\to k}| \\
& -\beta \sum_{k'\in N(i)\setminus j} \left(\prod_{m\in N(k')\setminus i} \operatorname{sgn}(\nu_{m\to k'}) \right) \min_{m'\in N(k')\setminus i} |\nu_{m'\to k'}| \Bigg| \\
\leq \max_{(i,j)\in\mathcal{E}} & \beta \sum_{k\in N(i)\setminus j} D_{i,k} \\
\leq \beta (d_{\mathrm{v}} - 1) & \max_{(i,k)\in\mathcal{E}} D_{i,k} \\
\leq \beta (d_{\mathrm{v}} - 1) & \max_{\substack{(i,k)\in\mathcal{E} \\ m\in N(k)\setminus i}} |\mu_{m\to k} - \nu_{m\to k}| \\
= \beta (d_{\mathrm{v}} - 1) & \|\boldsymbol{\mu} - \boldsymbol{\nu}\|_\infty .
\end{aligned}$$

This implies that $\mathtt{W}$ is a contraction on $\left(\mathbb{R}^{|\mathcal{E}|}, \|\cdot\|_\infty\right)$ if $\beta < \frac{1}{d_{\mathrm{v}}-1}$.

The second part of the theorem follows from the contraction mapping theorem. Therefore, for an arbitrary $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code and any $0 \leq \beta < \frac{1}{d_{\mathrm{v}}-1}$, the WMS algorithm converges to a unique fixed point, i.e., $\mu^{(\ell)}_{i\to j} \to \mu^{*}_{i\to j}$ and $\mu^{(\ell)}_{i\leftarrow j} \to \mu^{*}_{i\leftarrow j}$, as $\ell \to \infty$.

For any WMS-consistent fixed point, there are two ways to prove the ML optimality of the hard decisions. One way, as shown in our earlier work [74], is by looking at *Problem-P* directly. Therein, we generalize the definition of a *minimal T-local deviation* of [1] to $T \geq \frac{1}{4}\mathrm{girth}(\mathcal{G})$. By using the generalized *minimal T-local deviation*, we show that, if the fixed point is WMS-consistent, the corresponding hard-decision vector is a locally optimal codeword. By the fact that local optimality implies both global optimality and LP optimality [1], the hard-decision vector is an ML codeword, and also, an integer LP optimal solution. A summary of [74] is provided in Appendix B.1.

The other method, which is introduced in the rest of this section, is by solving *Problem-D*. When the WMS algorithm converges to a WMS-consistent fixed point, we construct a dual witness, denoted by $\boldsymbol{\tau}^* \in \mathbb{R}^{|\mathcal{E}|}$, using the method proposed in [66]. By Theorem 51, it can be shown that the vector $\boldsymbol{\tau}^*$ is a dual optimal point of *Problem-P*. Also, the hard-decision vector is an LP optimal codeword and, hence, an ML codeword.

The following lemma shows that the vector $\boldsymbol{\tau}^*$, which is constructed from the fixed-point messages $\mu^*_{i\to j}$ and $\mu^*_{i\leftarrow j}$, is a dual feasible point of *Problem-P*.

Lemma 47. *Consider the WMS algorithm with $\beta < \frac{1}{d_\mathrm{v}-1}$ for a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code. Let the bit node and check node messages, $\mu^*_{i\to j}$ and $\mu^*_{i\leftarrow j}$, be the unique fixed point of the WMS message update rule. The vector $\boldsymbol{\tau}^* \in \mathbb{R}^{|\mathcal{E}|}$, defined by*

$$\tau^*_{i,j} \triangleq \frac{1}{d_\mathrm{v}} \left(\mu^*_{i\to j} - \beta \left(d_\mathrm{v} - 1 \right) \mu^*_{i\leftarrow j} \right), \tag{3.31}$$

is a dual feasible point of Problem-P.

Proof. Fix a variable node $i \in \mathcal{V}_\mathrm{v}$. The sum of the dual variables on the edges incident to i is given by

$$\begin{aligned}
\sum_{j\in N(i)} \tau^*_{i,j} &= \frac{1}{d_\mathrm{v}} \sum_{j\in N(i)} \left(\mu^*_{i\to j} - \beta \left(d_\mathrm{v} - 1 \right) \mu^*_{i\leftarrow j} \right) \\
&= \frac{1}{d_\mathrm{v}} \sum_{j\in N(i)} \left(\mu^*_{i\to j} - \beta \sum_{k\in N(i)\setminus j} \mu^*_{i\leftarrow k} \right) \\
&= \gamma_i.
\end{aligned}$$

This proves the lemma.

Remark 48. In contrast to the construction in [66, Lemma 1], Lemma 47 is a simpli-

fied version that considers a one-step update of the WMS messages. In [66], min-sum messages over L iterations are considered. For a computation tree $\mathcal{T}_j^{2L}$ of depth $2L$ rooted at the check node j, those min-sum messages are used to generate an assignment $\boldsymbol{\tau}(j, L)$ to edges in $\mathcal{T}_j^{2L}$. Koetter and Vontobel showed that the dual feasible point $\boldsymbol{\tau}^*$ can be obtained by averaging $\boldsymbol{\tau}(j, L)$ over all $j \in \mathcal{V}_\mathrm{c}$. Since the number of leaf nodes in a computation tree increases exponentially in the depth of the computation tree, a weight factor α is introduced to attenuate the influence of the leaves of the computation tree. In our analysis, by the fact that the WMS messages satisfy a fixed-point equation, we simplify the construction and consider only the assignments on the computation tree $\mathcal{T}_j^2$ of depth 2. Next, we will show that the proposed dual feasible point $\boldsymbol{\tau}^*$ is also a optimal point in *Problem-D* if it is constructed from a WMS-consistent fixed point.

For a $j \in \mathcal{V}_\mathrm{c}$, let $\boldsymbol{\mu}_j^* \in \mathbb{R}^{d_\mathrm{c}}$ with $(\boldsymbol{\mu}_j^*)_m = \mu^*_{\varphi_j(m)\to j}$ for $m = 1, 2, \ldots, d_\mathrm{c}$ be a vector of the converged messages passed to j, where $\varphi_j(m)$ is defined in Section 3.2.5. Recall that $\boldsymbol{\tau}_j^* \in \mathbb{R}^{d_\mathrm{c}}$ is a vector with $(\boldsymbol{\tau}_j^*)_m = \tau^*_{\varphi_j(m),j}$. Let $\mathrm{sgn}(\boldsymbol{\mu}_j^*)$ be a vector which is composed of the signs of the entries in $\boldsymbol{\mu}_j^*$. The following lemma shows that an affine function of $\mathrm{sgn}(\boldsymbol{\mu}_j^*)$ minimizes the inner product $\langle \boldsymbol{w}, \boldsymbol{\tau}_j^* \rangle$ for all $\boldsymbol{w} \in \mathcal{L}$ when the fixed point is WMS-consistent. Recall that $\mathcal{L}$ is defined in (3.19). We use $\mathbf{1}$ to denote an all-one vector, and the vector dimension is determined by the context.

Lemma 49. *Consider a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code. If the WMS algorithm with $\beta < \frac{1}{d_\mathrm{v}-1}$ converges to a WMS-consistent fixed point $\mu^*_{i\to j}$ and $\mu^*_{i\leftarrow j}$ for all $(i, j) \in \mathcal{E}$, then for each $j \in \mathcal{V}_\mathrm{c}$,*

$$\arg\min_{\boldsymbol{w}\in\mathcal{L}} \left\langle \boldsymbol{w}, \boldsymbol{\tau}_j^* \right\rangle = \frac{1}{2}\left(\mathbf{1} - \mathrm{sgn}\left(\boldsymbol{\mu}_j^*\right)\right). \tag{3.32}$$

Proof. For a fixed $j \in \mathcal{V}_\mathrm{c}$, let $\overrightarrow{\boldsymbol{\mu}} \triangleq \boldsymbol{\mu}_j^*$, and $\overleftarrow{\boldsymbol{\mu}} \in \mathbb{R}^{d_\mathrm{c}}$ be a vector with $\overleftarrow{\mu}_m = \mu^*_{\varphi_j(m)\leftarrow j}$

for $m = 1, 2, \ldots, d_{\mathrm{c}}$. The inner product in (3.32) can be rewritten as

$$\begin{aligned}\left\langle \boldsymbol{w}, \boldsymbol{\tau}_j^* \right\rangle &= \sum_{m=1}^{d_{\mathrm{c}}} w_m \left(\boldsymbol{\tau}_j^*\right)_m \\ &\overset{\text{(a)}}{=} \frac{1}{d_{\mathrm{v}}} \sum_{m=1}^{d_{\mathrm{c}}} w_m \left(\vec{\mu}_m - \beta \left(d_{\mathrm{v}} - 1\right) \overleftarrow{\mu}_m\right) \\ &\overset{\text{(b)}}{=} \frac{1}{d_{\mathrm{v}}} \sum_{m=1}^{d_{\mathrm{c}}} w_m \operatorname{sgn}\left(\vec{\mu}_m\right) \left(\left|\vec{\mu}_m\right| - \beta \left(d_{\mathrm{v}} - 1\right) \left|\overleftarrow{\mu}_m\right|\right), \end{aligned} \tag{3.33}$$

where the equality (a) follows from (3.31), and the equality (b) holds because $\boldsymbol{\mu}_j^*$ is WMS-consistent. Let $|\vec{\mu}_{m_1}|$ and $|\vec{\mu}_{m_2}|$ be the two smallest absolute values in the vector $|\vec{\boldsymbol{\mu}}|$ and $|\vec{\mu}_{m_1}| \leq |\vec{\mu}_{m_2}|$. From the message update rule of the WMS algorithm (3.14), one can show that

$$\left|\overleftarrow{\mu}_m\right| = \begin{cases} \left|\vec{\mu}_{m_2}\right|, & \text{if } m = m_1, \\ \left|\vec{\mu}_{m_1}\right|, & \text{otherwise.} \end{cases}$$

Thus, the summation in (3.33) becomes

$$\begin{aligned}&\frac{1}{d_{\mathrm{v}}} w_{m_1} \operatorname{sgn}\left(\vec{\mu}_{m_1}\right) \left(\left|\vec{\mu}_{m_1}\right| - \beta \left(d_{\mathrm{v}} - 1\right) \left|\vec{\mu}_{m_2}\right|\right) \\ &\qquad + \frac{1}{d_{\mathrm{v}}} \sum_{m=1, m \neq m_1}^{d_{\mathrm{c}}} w_m \operatorname{sgn}\left(\vec{\mu}_m\right) \left(\left|\vec{\mu}_m\right| - \beta \left(d_{\mathrm{v}} - 1\right) \left|\vec{\mu}_{m_1}\right|\right). \end{aligned}$$

Since $0 < \beta \left(d_{\mathrm{v}} - 1\right) \leq 1$, one can show that

$$\left|\left|\vec{\mu}_{m_1}\right| - \beta \left(d_{\mathrm{v}} - 1\right) \left|\vec{\mu}_{m_2}\right|\right| \leq \left|\left|\vec{\mu}_m\right| - \beta \left(d_{\mathrm{v}} - 1\right) \left|\vec{\mu}_{m_1}\right|\right|$$

for all $m \neq m_1$. Thus, the minimum of the inner product in the LHS of (3.32) is

achieved by choosing

$$w_m = \begin{cases} \sum_{m'=1, m' \neq m_1}^{d_c} w_{m'} \mod 2 & \text{if } m = m_1, \\ \frac{1}{2}\left(1 - \operatorname{sgn}\left(\vec{\mu}_m\right)\right) & \text{otherwise.} \end{cases}$$

By Theorem 44, the vector $\frac{1}{2}\left(\mathbf{1} - \operatorname{sgn}\left(\boldsymbol{\mu}_j^*\right)\right)$ satisfies the j-th check node, and therefore we know that $w_{m_1} = \frac{1}{2}\left(1 - \operatorname{sgn}\left(\vec{\mu}_{m_1}\right)\right) = \frac{1}{2}\left(1 - \operatorname{sgn}\left(\left(\boldsymbol{\mu}_j^*\right)_{m_1}\right)\right)$. This completes the proof.

Remark 50. The proof of Lemma 49 employs an important observation in the proof of [66, Lemma 3]. Namely, given a check node, there are at least $d_c - 1$ outgoing messages with the same absolute value. If there is an outgoing message with a different absolute value, it will be along the edge where the incoming message with the smallest absolute value was passed, i.e., the edge $(\varphi_j(m_1), j)$. Also, we note that the absolute value of the m_1-th entry of $\boldsymbol{\tau}_j^*$ is the smallest among the absolute values of the entries of $\boldsymbol{\tau}_j^*$. With the goal of minimizing the inner product in (3.32), the entry of the binary vector $\boldsymbol{w}$ corresponding to the smallest absolute value of $\boldsymbol{\tau}_j^*$ has to be the modulo-2 sum of the other entries. Since the MS messages are not guaranteed to converge, Koetter and Vontobel computed the dual feasible point using computation trees of depth greater than one. Also, the assumption of large initial values for the MS algorithm is required for their construction of an optimal dual feasible point for *Problem-D*.

Similar to the idea in [66], we further find the ML optimal codeword after obtaining the local configuration for each $j \in \mathcal{V}_c$ based on $\boldsymbol{\tau}_j^*$. Let $\boldsymbol{\tau} \in \mathbb{R}^{|\mathcal{E}|}$ be a dual feasible point, and $g(\boldsymbol{\tau})$ be the objective function in *Problem-D*. Consider the dual feasible point $\boldsymbol{\tau}^*$ defined in (3.31), and let $\boldsymbol{w}_j \triangleq \frac{1}{2}(\mathbf{1} - \operatorname{sgn}(\boldsymbol{\mu}_j^*))$ be the local

assignment to check j, where $\boldsymbol{\mu}_j^*$ is the vector of converged messages passed to j. By Lemma 49, one can show that the objective function in *Problem-D* evaluated at $\boldsymbol{\tau}^*$ is

$$g(\boldsymbol{\tau}^*) = \sum_{j \in \mathcal{V}_c} \left\langle \boldsymbol{\tau}_j^*, \boldsymbol{w}_j \right\rangle .$$

However, to solve *Problem-D*, one needs to search over all $\boldsymbol{\tau}$ in the dual feasible set and find the maximum of $g(\boldsymbol{\tau})$. In the following theorem, we will show that $\boldsymbol{\tau}^*$ is actually an optimal solution of *Problem-D* if the WMS fixed point is WMS-consistent. This also implies that the corresponding hard decisions define an ML codeword.

Theorem 51. *Consider the WMS algorithm with $\beta < \frac{1}{d_v - 1}$. If the message vector $\boldsymbol{\mu}^{(\ell)}$ converges to a WMS-consistent fixed point, $\boldsymbol{\mu}^*$, then the vector of hard decisions $\boldsymbol{x}^* \in \{0,1\}^n$ with*

$$x_i^* = \frac{1}{2}\left(\mathbf{1} - \operatorname{sgn}\left(\gamma_i + \beta \sum_{j \in N(i)} \mu_{i \leftarrow j}^*\right)\right)$$

is a codeword. Also, $\boldsymbol{x}^*$ is LP optimal for Problem-P *and, hence, ML optimal.*

Proof. Let the optimal value of *Problem-P* and the optimal value of *Problem-D* be f^* and g^*, respectively. Since $\boldsymbol{\tau}^*$ proposed in Lemma 47 is in the feasible set of *Problem-D*, it is obvious that $g^* \geq g(\boldsymbol{\tau}^*)$. Let $\boldsymbol{w}_j \in \mathcal{L}$ be the binary vector that minimizes the inner product $\langle \boldsymbol{w}, \boldsymbol{\tau}_j^* \rangle$ over all $\boldsymbol{w} \in \mathcal{L}$. From Lemma 49, we know that $\boldsymbol{w}_j = \frac{1}{2}\left(\mathbf{1} - \operatorname{sgn}\left(\boldsymbol{\mu}_j^*\right)\right)$ for each $j \in \mathcal{V}_c$. Since $\boldsymbol{\mu}^*$ is WMS-consistent, we also know that $\left(\boldsymbol{w}_j\right)_m = x_{\varphi_j(m)}^*$ for $m = 1, 2, \ldots, d_c$. By using these facts, we will show that $\boldsymbol{x}^*$ is LP optimal by contradiction.

Assume that $\boldsymbol{x}^*$ does not minimize *Problem-P*, then we have

$$\begin{aligned} f^* &< \sum_{i\in\mathcal{V}_\mathrm{v}} \gamma_i x_i^* \overset{\text{(a)}}{=} \sum_{i\in\mathcal{V}_\mathrm{v}} \left(\sum_{j\in N(i)} \tau_{i,j}^* \right) x_i^* \\ &= \sum_{j\in\mathcal{V}_\mathrm{c}} \left(\sum_{i\in N(j)} \tau_{i,j}^* x_i^* \right) = \sum_{j\in\mathcal{V}_\mathrm{c}} \left(\sum_{m=1}^{d_\mathrm{c}} \tau_{\varphi_j(m),j}^* x_{\varphi_j(m)}^* \right) \\ &\overset{\text{(b)}}{=} \sum_{j\in\mathcal{V}_\mathrm{c}} \left\langle \boldsymbol{\tau}_j^*, \boldsymbol{w}_j \right\rangle = g\left(\boldsymbol{\tau}^*\right) \leq g^*, \end{aligned}$$

where (a) follows from Lemma 47, and (b) is a result of the WMS-consistency. However, weak duality implies that $f^* \geq g^*$. This gives a contradiction. Thus, $\boldsymbol{x}^*$ minimizes the primal problem, and hence, is LP optimal. Moreover, since $\boldsymbol{x}^* \in \mathcal{C}$, it is also an ML codeword.

Remark 52. Consider the WMS algorithm on a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code with $\beta < \frac{1}{d_\mathrm{v}-1}$. From Theorem 51, we are able to check the ML optimality of the WMS solution by testing the WMS-consistency conditions. Namely, if the messages satisfy the consistency conditions, then the hard decisions return an ML codeword.

Remark 53. Although the WMS algorithm and the AttMP algorithm are equivalent for a finite number of iterations, our results show that, for guaranteed convergence, the WMS algorithm allows a larger weight factor than the AttMP algorithm. To compare the convergence of these two algorithms, it is clear that the AttMP algorithm converges if and only if $\mu_{i\to j}^{(\ell)}(x) \to \mu_{i\to j}^*(x)$ and $\mu_{i\leftarrow j}^{(\ell)}(x) \to \mu_{i\leftarrow j}^*(x)$ for all $(i,j) \in \mathcal{E}$ and $x \in \{0,1\}$ as $\ell \to \infty$. On the other hand, when considering the convergence of the WMS algorithm, we first note that the WMS messages are simply the differences of the AttMP messages. That is, $\mu_{i\to j}^{(\ell)} = \mu_{i\to j}^{(\ell)}(0) - \mu_{i\to j}^{(\ell)}(1)$ and $\mu_{i\leftarrow j}^{(\ell)} = \mu_{i\leftarrow j}^{(\ell)}(0) - \mu_{i\leftarrow j}^{(\ell)}(1)$. The convergence of the WMS messages includes the following two cases in the underlying AttMP message space. First, if the underlying AttMP messages

converge, the corresponding WMS messages converge as well. For example, when the WMS algorithm has the weight factor $\beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$, we know that the WMS messages converge from Theorem 46. In fact, the underlying AttMP messages must converge according to Theorem 37. The second case is that the underlying AttMP messages do not converge, but their differences converge. For example, consider the case when $\frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)} \leq \beta < \frac{1}{d_{\mathrm{v}}-1}$ and $\gamma_i = \gamma > 0$ for all $i \in \mathcal{V}_{\mathrm{v}}$. One can show that the AttMP message $\mu_{i\to j}^{(\ell)}(0)$, defined in (3.10), is lower bounded by $(\ell+1)\gamma$ for any $(i,j) \in \mathcal{E}$. Thus, the AttMP message $\mu_{i\to j}^{(\ell)}(0)$ does not converge as $\ell \to \infty$. However, by Theorem 46, we know that the WMS message $\mu_{i\to j}^{(\ell)} \to \mu_{i\to j}^{*}$ as $\ell \to \infty$ for all $(i,j) \in \mathcal{E}$. This implies that the difference $\mu_{i\to j}^{(\ell)}(0) - \mu_{i\to j}^{(\ell)}(1)$ converges for all $(i,j) \in \mathcal{E}$. Since the convergence requirement for the WMS algorithm is weaker than the AttMP algorithm, the allowable weight factor for the convergence of the WMS algorithm is greater than the required weight factor for the AttMP algorithm.

3.4 Weighted Min-Sum Decoding with $\beta = \frac{1}{d_{\mathrm{v}}-1}$

We have shown that the unique fixed point of the WMS algorithm with $\beta < \frac{1}{d_{\mathrm{v}}-1}$ returns an ML codeword if the fixed point is WMS-consistent. In this section, the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ is considered. In this case, the WMS algorithm is not guaranteed to converge. Thus, we introduce a sufficient condition for the ML optimality of the hard-decision output when the WMS algorithm diverges. Moreover, when $\beta \geq \frac{1}{d_{\mathrm{v}}-1}$, the DE noise threshold of the WMS algorithm exists. We conjecture that the noise threshold of the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ gives a lower bound of the noise threshold of the LP decoder.

3.4.1 Optimality Guarantees

We begin this subsection by introducing notation and definitions for the following discussion. We denote the WMS messages $\{\mu_{i\to j}^{(\ell)} : (i,j) \in \mathcal{E}\}$ with $\beta = \frac{\delta}{d_{\mathrm{v}}-1}$ and

$\delta \in [0,1]$ in the ℓ-th iteration by a vector $\boldsymbol{\mu}_\delta^{(\ell)} \in \mathbb{R}^{|\mathcal{E}|}$, and a sequence of WMS message vectors $\{\boldsymbol{\mu}_\delta^{(\ell)} : \ell = 1, 2, \dots\}$ is denoted by $\{\boldsymbol{\mu}_\delta\}$. The hard decisions computed by $\boldsymbol{\mu}_\delta^{(\ell)}$ are denoted by $\boldsymbol{x}_\delta^{(\ell)} \in \{0,1\}^n$. When $\delta < 1$, Theorem 46 shows that the WMS algorithm converges to a fixed point $\boldsymbol{\mu}_\delta^* \in \mathbb{R}^{|\mathcal{E}|}$. The hard decisions computed using $\boldsymbol{\mu}_\delta^*$ are denoted by a vector $\boldsymbol{x}_\delta^* \in \{0,1\}^n$. For any WMS message vector $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{E}|}$, the vector $|\boldsymbol{\mu}| \in \mathbb{R}_+^{|\mathcal{E}|}$ consists of the absolute value of each element of $\boldsymbol{\mu}$. For any two WMS message vectors $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{E}|}$, we use the partial order $\boldsymbol{\mu} \succ \boldsymbol{\nu}$ to denote $\mu_{i\to j} > \nu_{i\to j}$ for all $(i,j) \in \mathcal{E}$, and $\boldsymbol{\mu} \succeq \boldsymbol{\nu}$ to denote $\mu_{i\to j} \geq \nu_{i\to j}$ for all $(i,j) \in \mathcal{E}$. We extend the definition of the WMS operator in (3.16) to $\mathtt{W}_\delta$ for $\beta = \frac{\delta}{d_\mathrm{v}-1}$ and $\delta \in [0,1]$.

The operator $\mathtt{W}_\delta$ preserves the partial order of the absolute values of the WMS messages under the following conditions.

Lemma 54. *Consider a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code and a particular LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$. Let $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{E}|}$ be two WMS-consistent message vectors. If $|\boldsymbol{\mu}| \succeq |\boldsymbol{\nu}| \succ \frac{\|\boldsymbol{\gamma}\|_\infty}{\delta}\mathbf{1}$ for a $\delta \in (0,1]$ and $\operatorname{sgn}(\boldsymbol{\mu}) = \operatorname{sgn}(\boldsymbol{\nu})$, then $|\mathtt{W}_\delta[\boldsymbol{\mu}]| \succeq |\mathtt{W}_\delta[\boldsymbol{\nu}]|$ and $\operatorname{sgn}(\mathtt{W}_\delta[\boldsymbol{\mu}]) = \operatorname{sgn}(\mathtt{W}_\delta[\boldsymbol{\nu}]) = \operatorname{sgn}(\boldsymbol{\mu})$.*

Proof. See Appendix A.8.

When $\beta = \frac{1}{d_\mathrm{v}-1}$, we have observed three types of trajectories for the WMS messages. They converge to a fixed point, oscillate in a limit cycle, or diverge to $\pm\infty$. In this section, we are interested in the case where the sequence of WMS message vectors, $\{\boldsymbol{\mu}_1\}$, is divergent and WMS-consistent. We formalize this case by the following definition.

Definition 55. A sequence of WMS message vectors, $\{\boldsymbol{\mu}\}$, is *divergent and consistent* if for any $c > 0$, there exists an $L(c) > 0$ such that whenever $\ell \geq L(c)$, the WMS message vector satisfies $|\boldsymbol{\mu}^{(\ell)}| \succeq c\mathbf{1}$ and $\boldsymbol{\mu}^{(\ell)}$ is WMS-consistent.

Lemma 56. *For a divergent and consistent sequence of WMS message vectors* $\{\boldsymbol{\mu}_1\}$*, there exists an* $\hat{L}$ *such that* $\boldsymbol{\mu}_1^{(\hat{L}+\ell)}$ *is WMS-consistent and* $\operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L}+\ell)}) = \operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L})})$ *for all* $\ell \geq 0$*.*

Proof. For a given received LLR vector $\boldsymbol{\gamma}$, let $\hat{L} = L(\|\boldsymbol{\gamma}\|_\infty + \epsilon)$, where $\epsilon > 0$. From Definition 55, we know that $\boldsymbol{\mu}_1^{(\hat{L}+\ell)}$ is WMS-consistent for all $\ell \geq 0$.

We prove the second part of the theorem by induction. The base case holds when $\ell = 0$. Then, we consider the inductive step. Suppose that $\operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L}+\ell)}) = \operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L})})$ for some $\ell \geq 0$. Since $|\boldsymbol{\mu}_1^{(\hat{L}+\ell)}| \succ \|\boldsymbol{\gamma}\|_\infty \mathbf{1}$, by Lemma 54, we know that

$$\operatorname{sgn}\left(\boldsymbol{\mu}_1^{(\hat{L}+\ell+1)}\right) = \operatorname{sgn}\left(\mathtt{W}_1\left[\boldsymbol{\mu}_1^{(\hat{L}+\ell)}\right]\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_1^{(\hat{L}+\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_1^{(\hat{L})}\right).$$

Thus, we conclude that $\operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L}+\ell)}) = \operatorname{sgn}(\boldsymbol{\mu}_1^{(\hat{L})})$ for all $\ell \geq 0$.

Corollary 57. *For a divergent and consistent sequence of WMS message vectors* $\{\boldsymbol{\mu}_1\}$*, there exists an* $\hat{L}$ *such that the hard-decision vectors* $\boldsymbol{x}^{\hat{L}+\ell}$ *satisfy* $\boldsymbol{x}^{\hat{L}+\ell} = \boldsymbol{x}^{\hat{L}}$ *for all* $\ell \geq 0$*.*

Proof. The result follows from Lemma 56 and the hard decision equation defined in (3.15).

Given two positive integers $L_1 > L_0$, to simplify notation, let $I(L_0, L_1) \triangleq \{L_0, L_0+1, \ldots, L_1\}$ be the set of all integers from L_0 to L_1. A property of a sequence of WMS message vectors, $\{\boldsymbol{\mu}_\delta\}$, is introduced in the following definition.

Definition 58 (Block-wise monotone property)**.** A sequence of WMS message vectors $\{\boldsymbol{\mu}_\delta\}$ with $\delta \in (0, 1]$ is said to have the *block-wise monotone property* in $I(L_0, L_1)$, denoted by $\mathrm{BMP}(I(L_0, L_1))$, if for all $\ell \in I(L_0, L_1)$, 1) $\boldsymbol{\mu}_\delta^{(\ell)}$ is WMS-consistent, 2) $\operatorname{sgn}(\boldsymbol{\mu}_\delta^{(\ell)}) = \operatorname{sgn}(\boldsymbol{\mu}_\delta^{(L_0)})$, 3) $|\boldsymbol{\mu}_\delta^{(\ell)}| \succ \frac{\|\boldsymbol{\gamma}\|_\infty}{\delta}\mathbf{1}$, and 4) $|\boldsymbol{\mu}_\delta^{(L_1)}| \succeq |\boldsymbol{\mu}_\delta^{(L_0)}|$.

For any non-negative integer $k \geq 0$, let

$$I_k(L_0, L_1) \triangleq I\left(L_0 + k(L_1 - L_0),\, L_1 + k(L_1 - L_0)\right).$$

In the following analysis, we show that, if there is a pair of integers $L_1 > L_0 > 0$ such that the sequence of WMS message vectors, $\{\boldsymbol{\mu}_\delta\}$, satisfies BMP$(I_0(L_0, L_1))$, then $\{\boldsymbol{\mu}_\delta\}$ also satisfies BMP$(I_k(L_0, L_1))$ for all $k \geq 0$. We first show that if $\{\boldsymbol{\mu}_\delta\}$ satisfies BMP$(I_0(L_0, L_1))$, then $\{\boldsymbol{\mu}_\delta\}$ also satisfies BMP$(I_1(L_0, L_1))$.

Lemma 59. *Let $\boldsymbol{\gamma} \in \mathbb{R}^n$ be the received LLR vector, and $\{\boldsymbol{\mu}_\delta\}$ be the sequence of WMS message vectors of a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code with $\delta \in (0, 1]$. Suppose there exist $0 < L_0 < L_1$ such that $\{\boldsymbol{\mu}_\delta\}$ satisfies* BMP$(I_0(L_0, L_1))$*, then*

$$\operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(L_1+\ell')}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(L_0+\ell')}\right) \tag{3.34}$$

and

$$\left|\boldsymbol{\mu}_\delta^{(L_1+\ell')}\right| \succeq \left|\boldsymbol{\mu}_\delta^{(L_0+\ell')}\right| \tag{3.35}$$

for all $\ell' = 0, 1, \ldots, L_1 - L_0$.

Proof. We prove this lemma by induction. The base case, $\ell' = 0$, holds because conditions 2) and 4) of BMP$(I_1(L_0, L_1))$ are satisfied.

For the inductive step, suppose that $\operatorname{sgn}(\boldsymbol{\mu}_\delta^{(L_1+\ell')}) = \operatorname{sgn}(\boldsymbol{\mu}_\delta^{(L_0+\ell')})$ and $|\boldsymbol{\mu}_\delta^{(L_1+\ell')}| \succeq |\boldsymbol{\mu}_\delta^{(L_0+\ell')}|$ for some $\ell' > 0$. Since $\boldsymbol{\mu}_\delta^{(L_0+\ell')}$ satisfies conditions 1) and 3) of BMP$(I_0(L_0, L_1))$, from Lemma 54, we have

$$\left|\boldsymbol{\mu}_\delta^{(L_1+\ell'+1)}\right| \succeq \left|\boldsymbol{\mu}_\delta^{(L_0+\ell'+1)}\right|$$

and

$$\operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(L_1+\ell'+1)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(L_0+\ell'+1)}\right).$$

Since both the base case and the inductive step are proved, we know that (3.34) and (3.35) hold for $0 \le \ell' \le L_1 - L_0$.

Corollary 60. *Let $\boldsymbol{\gamma} \in \mathbb{R}^n$ be the received LLR vector, and $\{\boldsymbol{\mu}_\delta\}$ be the sequence of WMS message vectors of a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code with $\delta \in (0, 1]$. Suppose there exist $0 < L_0 < L_1$ such that $\{\boldsymbol{\mu}_\delta\}$ satisfies* $\mathrm{BMP}(I_0(L_0, L_1))$. *Then $\{\boldsymbol{\mu}_\delta\}$ also satisfies* $\mathrm{BMP}(I_1(L_0, L_1))$.

Proof. We first show that, for any $0 \le \ell' \le (L_1 - L_0)$, $\boldsymbol{\mu}_\delta^{(L_1+\ell')}$ is WMS-consistent. For a fixed $i \in \mathcal{V}_\mathrm{v}$, let $\mu_{\delta,i\leftarrow j}^{(L_1+\ell')}$ be the WMS message with $\beta = \frac{\delta}{d_\mathrm{v}-1}$ passed from $j \in N(i)$ to i in the $(L_1+\ell')$-th iteration. From the WMS message update rule (3.14), we know that, for any $j \in N(i)$,

$$\begin{aligned}\operatorname{sgn}\left(\mu_{\delta,i\leftarrow j}^{(L_1+\ell')}\right) &= \prod_{m\in N(j)\setminus i} \operatorname{sgn}\left(\mu_{\delta,m\to j}^{(L_1+\ell')}\right) \overset{\text{(a)}}{=} \prod_{m\in N(j)\setminus i} \operatorname{sgn}\left(\mu_{\delta,m\to j}^{(L_0+\ell')}\right) \\ &= \operatorname{sgn}\left(\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right) = \operatorname{sgn}\left(\mu_{\delta,i\to j}^{(L_0+\ell')}\right) \overset{\text{(b)}}{=} \operatorname{sgn}\left(\mu_{\delta,i\to j}^{(L_1+\ell')}\right), \end{aligned} \tag{3.36}$$

where the equalities (a) and (b) follow from (3.34). Since $\boldsymbol{\mu}_\delta^{(L_0+\ell')}$ is WMS-consistent, we know that

$$\begin{aligned}\operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right) &= \operatorname{sgn}\left(\gamma_i + \beta \sum_{j\in N(i)} \mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right) \\ &= \operatorname{sgn}\left(\gamma_i + \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right) \beta \sum_{j\in N(i)} \left|\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right|\right)\end{aligned}$$

for any $j_0 \in N(i)$. Therefore, one can show

$$\begin{aligned} 0 &< \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right)\left(\gamma_i + \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right)\beta \sum_{j\in N(i)}\left|\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right|\right) \\ &= \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right)\gamma_i + \beta \sum_{j\in N(i)}\left|\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right|. \end{aligned}$$

From the WMS message update rule (3.14) and Lemma 59, one can also see that $|\mu_{\delta,i\leftarrow j}^{(L_1+\ell')}| \geq |\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}|$ for all $j \in N(i)$. Thus, we have

$$\gamma_i + \beta \sum_{j'\in N(i)} \mu_{\delta,i\leftarrow j'}^{(L_1+\ell')} \neq 0 \tag{3.37}$$

and

$$\operatorname{sgn}\left(\gamma_i + \beta \sum_{j'\in N(i)} \mu_{\delta,i\leftarrow j'}^{(L_1+\ell')}\right) = \operatorname{sgn}\left(\mu_{\delta,i\to j}^{(L_1+\ell')}\right) \tag{3.38}$$

because

$$\begin{aligned} 0 &< \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_0+\ell')}\right)\gamma_i + \beta \sum_{j\in N(i)}\left|\mu_{\delta,i\leftarrow j}^{(L_0+\ell')}\right| \\ &\leq \operatorname{sgn}\left(\mu_{\delta,i\to j_0}^{(L_1+\ell')}\right)\gamma_i + \beta \sum_{j\in N(i)}\left|\mu_{\delta,i\leftarrow j}^{(L_1+\ell')}\right|. \end{aligned}$$

From (3.36), (3.37) and (3.38), we conclude that $\boldsymbol{\mu}_\delta^{(L_1+\ell')}$ is WMS-consistent.

Equation (3.34) implies that

$$\operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(L_1+\ell')}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(L_0+\ell')}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(L_1)}\right),$$

where the second equality follows from the condition 2) of BMP($I_0(L_0, L_1)$). Also from Lemma 59, we know that $|\boldsymbol{\mu}_\delta^{(\ell'+L_1)}| \succeq |\boldsymbol{\mu}_\delta^{(\ell'+L_0)}| \succ \frac{\|\boldsymbol{\gamma}\|_\infty}{\delta}\mathbf{1}$ for all $\ell' = 0, 1, \ldots, (L_1-$

L_0). Finally, by (3.35), we have

$$\left|\boldsymbol{\mu}_\delta^{(2L_1-L_0)}\right| \succeq \left|\boldsymbol{\mu}_\delta^{(L_1)}\right|.$$

Therefore, we conclude that $\{\boldsymbol{\mu}_\delta\}$ also satisfies $\mathrm{BMP}(I_1(L_0, L_1))$.

Remark 61. Although Corollary 60 shows that $\mathrm{BMP}(I_0(L_0, L_1))$ implies $\mathrm{BMP}(I_1(L_0, L_1))$, it can be easily extended to the statement that $\mathrm{BMP}(I_k(L_0, L_1))$ implies $\mathrm{BMP}(I_{k+1}(L_0, L_1))$ for any $k > 0$ by the same argument in the proof of Corollary 60.

Now, we extend the property to intervals $I_k(L_0, L_1)$ for all $k \geq 0$.

Lemma 62. *Let $\boldsymbol{\gamma} \in \mathbb{R}^n$ be the received LLR vector, and $\{\boldsymbol{\mu}_\delta\}$ be the sequence of WMS message vector of a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code with $\delta \in (0, 1]$. Suppose there exist $0 < L_0 < L_1$ such that the sequence of WMS message vectors, $\{\boldsymbol{\mu}_\delta\}$, satisfies $\mathrm{BMP}(I_0(L_0, L_1))$. Then, for all $\ell \geq L_0$,*

$$\mathrm{sgn}\left(\boldsymbol{\mu}_\delta^{(\ell)}\right) = \mathrm{sgn}\left(\boldsymbol{\mu}_\delta^{(L_0)}\right)$$

and

$$\left|\boldsymbol{\mu}_\delta^{(\ell)}\right| \succ \frac{\|\boldsymbol{\gamma}\|_\infty}{\delta}\mathbf{1}.$$

Proof. Since that the set of integers greater or equal to L_0 can be written as

$$\lim_{K\to\infty} \bigcup_{k=0}^{K} I_k(L_0, L_1),$$

the lemma can be proved by showing that $\{\boldsymbol{\mu}_\delta\}$ satisfies $\mathrm{BMP}(I_k(L_0, L_1))$ for any $k \geq 0$. We will prove this statement by induction.

The base case is obtained from the assumption when setting $k = 0$. Next, we consider the inductive step. Suppose that $\{\boldsymbol{\mu}_\delta\}$ satisfies BMP($I_k(L_0, L_1)$). From Corollary 60, we know that $\{\boldsymbol{\mu}_\delta\}$ also satisfies BMP($I_{k+1}(L_0, L_1)$). Thus, we know that $\boldsymbol{\mu}_\delta^{(\ell)}$ satisfies BMP($I_k(L_0, L_1)$) for any $k \geq 0$.

In the following analysis, we show that there exist $0 < L_0 < L_1$ and a $\delta \in (0, 1)$ such that $\{\boldsymbol{\mu}_\delta\}$ satisfies BMP($I_0(L_0, L_1)$) when $\{\boldsymbol{\mu}_1\}$ is divergent and consistent. We first show that, for any integer $L > 0$, the WMS message $\boldsymbol{\mu}_1^{(\ell)}$ for $\ell \leq L$ can be approximated by $\{\boldsymbol{\mu}_\delta\}$ with δ close enough to 1.

Lemma 63. *Consider a $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC code. Given the LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$, let $\{\boldsymbol{\mu}_1\}$ and $\{\boldsymbol{\mu}_\delta\}$ be two sequences of WMS message vectors with $\beta = \frac{1}{d_\mathrm{v}-1}$ and $\beta = \frac{\delta}{d_\mathrm{v}-1}$, respectively. For any $\epsilon > 0$ and integer $L > 0$, there exists a $\delta_0(L) > 0$ such that $\|\boldsymbol{\mu}_1^{(\ell)} - \boldsymbol{\mu}_\delta^{(\ell)}\|_\infty \leq \epsilon$ for all $0 \leq \ell \leq L$ whenever $\delta \in [\delta_0(L), 1]$.*

Proof. See Appendix A.9.

The following lemma shows the existence of $0 < L_0 < L_1$ and $\delta \in (0, 1]$ such that the sequence of WMS message vectors $\{\boldsymbol{\mu}_\delta\}$ satisfies BMP($I_0(L_0, L_1)$) when the sequence of WMS vectors $\{\boldsymbol{\mu}_1\}$ is divergent and consistent. Note that the choices of δ, L_0, and L_1 are also suggested in the proof of Lemma 63.

Lemma 64. *Given the received LLR vector, $\boldsymbol{\gamma} \in \mathbb{R}^n$, suppose that $\{\boldsymbol{\mu}_1\}$ is divergent and consistent. There exist L_0 and L_1 with $0 < L_0 < L_1$ and a $\delta \in (0, 1]$ such that $\{\boldsymbol{\mu}_\delta\}$ satisfies* BMP($I_0(L_0, L_1)$).

Proof. Since $\{\boldsymbol{\mu}_1\}$ is divergent and consistent, by Definition 55 and Lemma 56, we can choose $L_0 > 0$ such that, for all $\ell \geq L_0$: $\boldsymbol{\mu}_1^{(\ell)}$ is WMS-consistent; $|\boldsymbol{\mu}_1^{(\ell)}| \succeq 2\|\boldsymbol{\gamma}\|_\infty \mathbf{1}$; and $\operatorname{sgn}(\boldsymbol{\mu}_1^{(\ell)}) = \operatorname{sgn}(\boldsymbol{\mu}_1^{(L_0)})$ for all $\ell \geq L_0$. Similarly, we can also find an $L_1 > L_0$ such that

$$\left|\boldsymbol{\mu}_1^{(\ell)}\right| \succeq \left(\left\|\boldsymbol{\mu}_1^{(L_0)}\right\|_\infty + 2\|\boldsymbol{\gamma}\|_\infty\right)\mathbf{1}, \tag{3.39}$$

whenever $\ell \geq L_1$. From the proof of Lemma 63, we can choose $\epsilon = \frac{1}{2}\|\boldsymbol{\gamma}\|_\infty$ and

$$\delta \geq 1 - \frac{2\epsilon}{L_1 (L_1 + 1) \|\boldsymbol{\gamma}\|_\infty} = 1 - \frac{1}{L_1 (L_1 + 1)} \tag{3.40}$$

so that, for all $\ell \in I_0(L_0, L_1)$,

$$\left\| \boldsymbol{\mu}_1^{(\ell)} - \boldsymbol{\mu}_\delta^{(\ell)} \right\|_\infty \leq \epsilon = \frac{1}{2}\|\boldsymbol{\gamma}\|_\infty. \tag{3.41}$$

Note that Equation (3.40) and the inequalities $L_1 > L_0 > 2$ imply $\delta \geq \frac{19}{20}$. With these choices of L_0 and L_1, it can be shown that

$$\begin{aligned}
\left| \boldsymbol{\mu}_\delta^{(L_1)} \right| &\overset{\text{(a)}}{\preceq} \left(\|\boldsymbol{\mu}_1^{(L_0)}\|_\infty + 2\|\boldsymbol{\gamma}\|_\infty - \epsilon \right) \mathbf{1} \\
&= \left(\|\boldsymbol{\mu}_1^{(L_0)}\|_\infty + \frac{3}{2}\|\boldsymbol{\gamma}\|_\infty \right) \mathbf{1} \\
&\succ \left(\|\boldsymbol{\mu}_1^{(L_0)}\|_\infty + \frac{1}{2}\|\boldsymbol{\gamma}\|_\infty \right) \mathbf{1} \\
&= \left(\|\boldsymbol{\mu}_1^{(L_0)}\|_\infty + \epsilon \right) \mathbf{1} \\
&\succeq \left| \boldsymbol{\mu}_\delta^{(L_0)} \right|,
\end{aligned} \tag{3.42}$$

where the inequality (a) follows from (3.39) and (3.41). From (3.41), one can show, for all $\ell \in I_0(L_0, L_1)$,

$$\boldsymbol{\mu}_1^{(\ell)} - \frac{\mathbf{1}}{2}\|\boldsymbol{\gamma}\|_\infty \preceq \boldsymbol{\mu}_\delta^{(\ell)} \preceq \boldsymbol{\mu}_1^{(\ell)} + \frac{\mathbf{1}}{2}\|\boldsymbol{\gamma}\|_\infty.$$

Since $|\boldsymbol{\mu}_1^{(\ell)}| \succeq 2\|\boldsymbol{\gamma}\|_\infty \mathbf{1}$ for all $\ell \in I_0(L_0, L_1)$, one can have

$$\operatorname{sgn}\left(\boldsymbol{\mu}_\delta^{(\ell)} \right) = \operatorname{sgn}\left(\boldsymbol{\mu}_1^{(\ell)} \right)$$

and

$$\left|\boldsymbol{\mu}_{\delta}^{(\ell)}\right| \succeq \left|\boldsymbol{\mu}_{1}^{(\ell)}\right| - \epsilon\mathbf{1} \succeq \frac{3}{2}\|\boldsymbol{\gamma}\|_{\infty}\mathbf{1} \succ \frac{\|\boldsymbol{\gamma}\|_{\infty}}{\delta}\mathbf{1} \tag{3.43}$$

for all $\ell \in I_0(L_0, L_1)$. Since $\operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(L_0)}\right)$, we know that

$$\operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(L_0)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(L_0)}\right) \tag{3.44}$$

for all $\ell \in I_0(L_0, L_1)$. Since $\boldsymbol{\mu}_{1}^{(L_0)}$ is WMS-consistent, Equation (3.44) implies that $\boldsymbol{\mu}_{\delta}^{(\ell)}$ is WMS-consistent for all $\ell \in I_0(L_0, L_1)$ as well. By (3.42)–(3.44) and the fact that $\boldsymbol{\mu}_{\delta}^{(\ell)}$ is WMS-consistent for all $\ell \in I_0(L_0, L_1)$, we conclude that $\{\boldsymbol{\mu}_{\delta}\}$ satisfies $\mathrm{BMP}(I_0(L_0, L_1))$.

Corollary 65. *Given the received LLR vector, $\boldsymbol{\gamma} \in \mathbb{R}^n$, suppose that $\{\boldsymbol{\mu}_1\}$ is divergent and consistent. Then, there exist an $L_0 > 0$ and a $0 < \delta < 1$ such that, whenever $\ell \geq L_0$,*

$$\left|\boldsymbol{\mu}_{\delta}^{(\ell)}\right| \succeq \frac{\|\boldsymbol{\gamma}\|_{\infty}}{\delta}\mathbf{1} \tag{3.45}$$

and

$$\operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(L_0)}\right). \tag{3.46}$$

Proof. From Lemma 62 and Lemma 64, we obtain (3.45) and (3.46) directly.

Theorem 66. *Consider a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code and a particular LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$. Assume that the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ is divergent and consistent. There exists an $L > 0$ so that the hard decision vectors satisfy $\boldsymbol{x}^{(\ell)} = \boldsymbol{x}^{(L)}$ for all $\ell \geq L$, and $\boldsymbol{x}^{(L)}$ is an ML codeword.*

Proof. To prove this theorem, we show that there is a $\delta \in (0, 1)$ such that the WMS algorithm with $\beta = \frac{\delta}{d_{\mathrm{v}}-1}$ converges to a WMS-consistent fixed point whose hard decisions give the same codeword as $\boldsymbol{x}^{(L)}$.

From Corollary 65, there exist an $L_0 > 0$ and a $\delta \in (0,1)$ such that $\operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(\ell)}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(L_0)}\right)$ for all $\ell \geq L_0$. Since $\frac{\delta}{d_{\mathrm{v}}-1} < \frac{1}{d_{\mathrm{v}}-1}$, the message vector converges to a fixed point $\boldsymbol{\mu}_{\delta}^{*}$ and $\operatorname{sgn}\left(\boldsymbol{\mu}_{\delta}^{*}\right) = \operatorname{sgn}\left(\boldsymbol{\mu}_{1}^{(L_0)}\right)$. Since $\boldsymbol{\mu}_{1}^{(L_0)}$ is WMS-consistent, the converged message vector $\boldsymbol{\mu}_{\delta}^{*}$ is also WMS-consistent. Hence, for all $(i,j) \in \mathcal{E}$

$$\begin{aligned}\operatorname{sgn}\left(\gamma_i + \beta \sum_{j \in N(i)} \mu_{\delta,i\leftarrow j}^{*}\right) &= \operatorname{sgn}\left(\mu_{\delta,i\to j}^{*}\right) \\ &= \operatorname{sgn}\left(\mu_{1,i\to j}^{(L_0)}\right) = \operatorname{sgn}\left(\gamma_i + \frac{1}{d_{\mathrm{v}}-1} \sum_{j \in N(i)} \mu_{1,i\leftarrow j}^{(L_0)}\right).\end{aligned}$$

For any $i \in \mathcal{V}_{\mathrm{v}}$, the hard decision $x_{\delta,i}^{*}$ with $\beta = \frac{\delta}{d_{\mathrm{v}}-1}$ is

$$\begin{aligned}x_{\delta,i}^{*} &= \frac{1}{2}\left(1 - \operatorname{sgn}\left(\gamma_i + \beta \sum_{j \in N(i)} \mu_{\delta,i\leftarrow j}^{*}\right)\right) \\ &= \frac{1}{2}\left(1 - \operatorname{sgn}\left(\gamma_i + \frac{1}{d_{\mathrm{v}}-1} \sum_{j \in N(i)} \mu_{1,i\leftarrow j}^{(L_0)}\right)\right) \\ &= x_i^{(L_0)}.\end{aligned}$$

From Theorem 51, we know that $\boldsymbol{x}_{\delta}^{*}$ is LP and ML optimal. Therefore, the hard decision vector $\boldsymbol{x}^{(L_0)}$ is also an LP and ML optimal codeword. By setting $L = L_0$ and from Corollary 57, we conclude the proof.

Remark 67. In this section, we considered the WMS algorithm as a DP problem with discount factor $\beta(d_{\mathrm{v}} - 1) \leq 1$. When $\beta = \frac{1}{d_{\mathrm{v}}-1}$ and the sequence of WMS message vectors $\{\boldsymbol{\mu}\}$ is divergent and consistent, the WMS update is equivalent to a Markov decision process (MDP) problem with discount factor 1. Theorem 66 essentially states that WMS decoding always has the natural analog of a Blackwell optimal

policy [75] if $\{\boldsymbol{\mu}\}$ is divergent and consistent according to Definition 55.

3.4.2 Connections with LP Thresholds

In this subsection, we connect the LP threshold estimation with both the WMS algorithm and the DE-type analysis in [1, 67]. We have shown that when the WMS algorithm with $\beta < \frac{1}{d_{\mathrm{v}}-1}$ converges to a set of WMS-consistent messages, the WMS algorithm returns a codeword which is LP optimal. Similarly, when the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ is divergent and consistent, the WMS algorithm also returns a codeword which is LP optimal. If the following conjecture is true, we can conclude that the threshold of the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ gives a lower bound for the threshold of LP decoding.

Conjecture 68. *Consider WMS decoding of $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes with girth $\Omega(\log n)$ over the BSC with cross-over probability p and let p^* be the bit-error rate threshold for WMS decoding with $\beta = \frac{1}{d_{\mathrm{v}}-1}$. Then, as $n \to \infty$, WMS decoding diverges to consistent messages with probability $1 - o_n(1)$ for all $p < p^*$.*

Remark 69. The DE analysis of the WMS decoding of $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ gives automatically that almost all messages diverge to consistent values (i.e., a BER threshold). Conjecture 68 is that p^* is also a word-error rate (WER) threshold. Conjecture 68 has been tested via simulation, and we are currently pursuing a rigorous proof.

Example 70. Consider a $(3, 6)$-regular LDPC code over the BSC. From a DE analysis of the WMS algorithm (i.e., not the DE-type analysis for local optimality as proposed in [1]) with $\beta = 1/2$, one finds that the WMS algorithm will decode correctly when $p \leq 0.055$. Note that the obtained LP threshold lower bound matches the best possible bound using techniques from [1].

3.5 Numerical Results

The WER for the WMS algorithms and the probability of not converging to a set of consistent messages are shown in Figure 3.1. The solid lines are the WER of the WMS algorithm, and the dashed lines are the probability of *not* satisfying WMS-consistent. The simulation is conducted over a $(3, 6)$-regular LDPC code ensemble with $n = 10^4$. Two weight factors, $\beta = 0.49$ and $\beta = 0.5$, are considered, and a maximum of 500 iterations are used to decode each codeword. Both the BSC and the BIAWGNC are tested. As shown in Figure 3.1, when $\beta = 0.49$, the WMS algorithm may converge to a set of messages that are not WMS-consistent even though the codeword is successfully decoded. However, when $\beta = 0.5$, those two probabilities become nearly identical.

In [1] and [67], a DE-type analysis is employed to compute the threshold for having the all-zero codeword as a locally optimal codeword. Whenever the channel noise is below this threshold, the probability that the all-zero codeword is not a locally optimal codeword goes to 0 as the number of iterations goes to infinity. Since the local optimality implies the LP and global optimality, this threshold is a lower bound of the LP decoding threshold. In Figure 3.2, we plot the thresholds obtained by DE-type analysis versus some weight factors β. It is worth noting that according to our numerical results, the best lower bound for the LP decoding threshold, in all cases, are obtained when $\beta = \frac{1}{d_v - 1}$. When $\beta < \frac{1}{d_v - 1}$, the DE-type analysis proposed in [1] has no threshold effect. The threshold effect does not occur because the density of the correlation between the best skinny trees and the channel output in [1, 67] converges to a fixed point instead of diverging to $\pm\infty$. Therefore, the probability of not having the all-zero codeword as a locally optimal codeword can not be arbitrarily small. Note that the weight vector we used in Figure 3.2 is

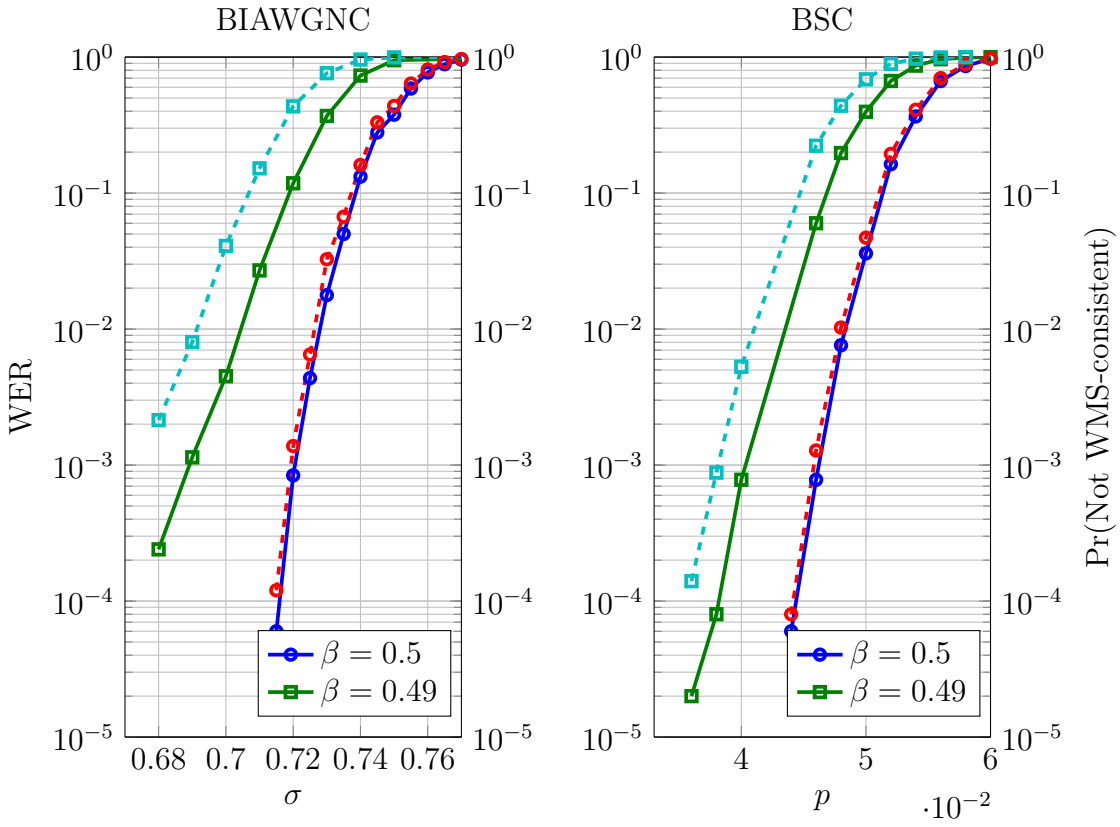

Figure 3.1: The WER (solid lines) of the WMS algorithm for $(3, 6)$-regular LDPC code and the probability of converging to a set of messages that are not WMS-consistent (dashed lines).

different from the weight factor proposed in [1]. In Figure 3.2, only one weight factor $\beta \in [0, 1]$ is employed. For a T-depth skinny tree, this is equivalent to weighting the tree from the root to the leafs by a weight vector $(1, \beta, \beta^2, \cdots, \beta^T)$. In [1], Arora *et al.* considered a two-stage weighting strategy. For a fixed $s < T$, the weight vector is $(\rho, \rho, \ldots, \rho, 1, \beta, \beta^2, \cdots, \beta^s) \in \mathbb{R}^T$. In the first $s < T$ iterations, exponential weights are employed. According to the density of the correlation between the best skinny trees and the channel output, a proper ρ is chosen so that the threshold effect can occur. Although this two-stage weighting strategy in [1] has a threshold effect for any $\beta \in [0, 1]$, the observed best threshold also occurs when $\beta = \frac{1}{d_v - 1}$.

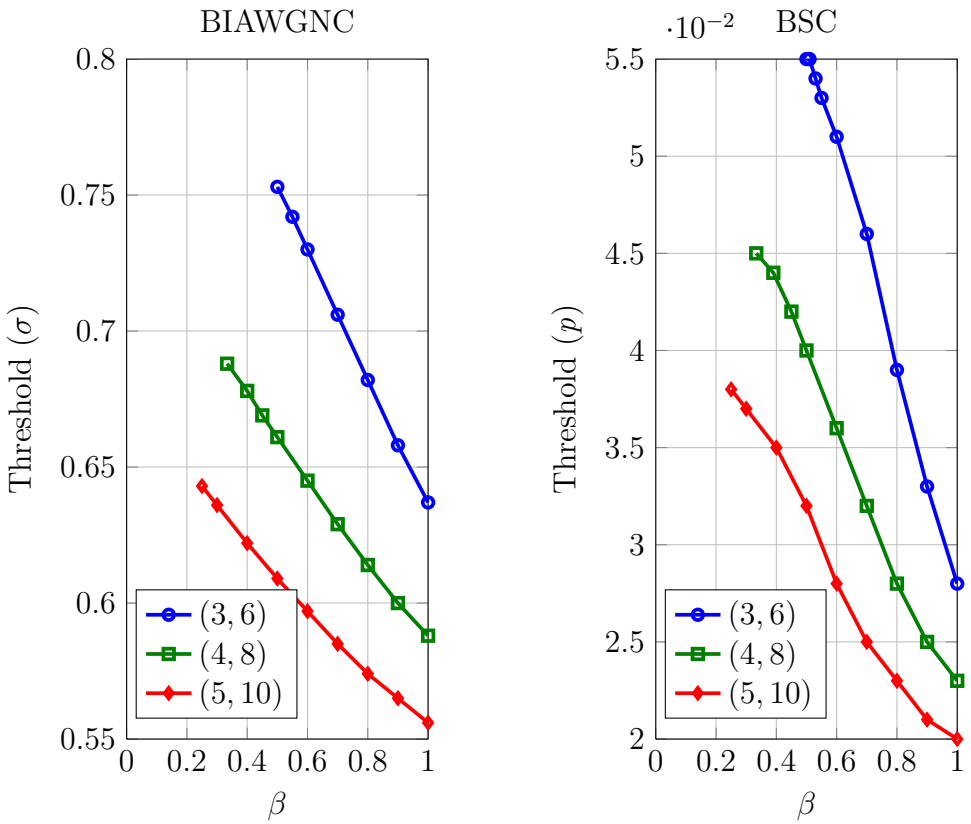

Figure 3.2: The lower bound of the LP decoding threshold for $(3, 6)$, $(4, 8)$ and $(5, 10)$-regular LDPC codes over the BIAWGNC and the BSC.

The comparisons of the WER performance between the WMS algorithm and the TRMP algorithm [59] are shown in Figure 3.3. For any strictly positive pairwise Markov random field (MRF) with binary variables, it has been shown that the fixed point of the TRMP algorithm always specifies an optimal LP dual solution [59, 76]. The TRMP message update rules in the logarithmic domain are

$$\nu_{i\to j}^{(\ell+1)} = \gamma_i + \rho \sum_{k\in N(i)\setminus j} \nu_{i\leftarrow k}^{(\ell)} - (1-\rho)\,\nu_{i\leftarrow j}^{(\ell)},$$

$$\nu_{i\leftarrow j}^{(\ell+1)} = \rho \left(\prod_{m\in N(j)\setminus i} \operatorname{sgn}\left(\nu_{m\to j}^{(\ell)}\right) \right) \min_{m'\in N(j)\setminus i} \left|\nu_{m'\to j}^{(\ell)}\right| - (1-\rho)\,\nu_{i\to j}^{(\ell)},$$

where $\rho \leq 1$ is the edge appearance probability in a random spanning tree. An uniform edge appearance probability $\rho = \frac{n(1+d_v/d_c)-1}{|\mathcal{E}|}$ is employed in our simulation. One can notice that these message update rules are similar to the WMS algorithms. Although, the factors associated with the factor graph of an LDPC code are not strictly positive, the optimality of the TRMP hard decisions is observed in a numerical simulation of a $(3, 4)$-regular LDPC code with $n = 12$. Thus, we take the TRMP algorithm into consideration, and compare its WER performance with the WER performance of the WMS algorithms.

In this comparison, a $(3, 6)$-regular LDPC codes over the BSC is considered, and the codeword length for both algorithms is $n = 10^4$. Three weight factors for the WMS algorithm are tested: $\beta = 0.5$, which is discussed in this section; $\beta = 0.8$, which has been shown to have the best performance by DE analysis [70]; and $\beta = 1$, which is equivalent to the MS algorithm for LDPC codes. All WMS algorithms perform 100 iterations in decoding a codeword. In the case of the TRMP algorithm, two simulations with 100 iterations and 1000 iterations, respectively, for decoding a codeword are conducted. As shown in Figure 3.3, the WER performance of the TRMP algorithm with 1000 iterations is close to the WMS algorithm with $\beta = 1$. However, if the TRMP algorithm only performs 100 iterations in decoding each codeword, it becomes close to the WMS algorithm with $\beta = 0.5$. The performance loss of the TRMP algorithm with 100 iterations is caused by the insufficient number of iterations. Since the TRMP algorithm is not close enough to the converged point, the corresponding hard decisions are not reliable. Although the TRMP algorithm over the binary alphabet has been shown to be LP optimal when the algorithm converges, finding the noise threshold of the TRMP algorithm is still an open problem.

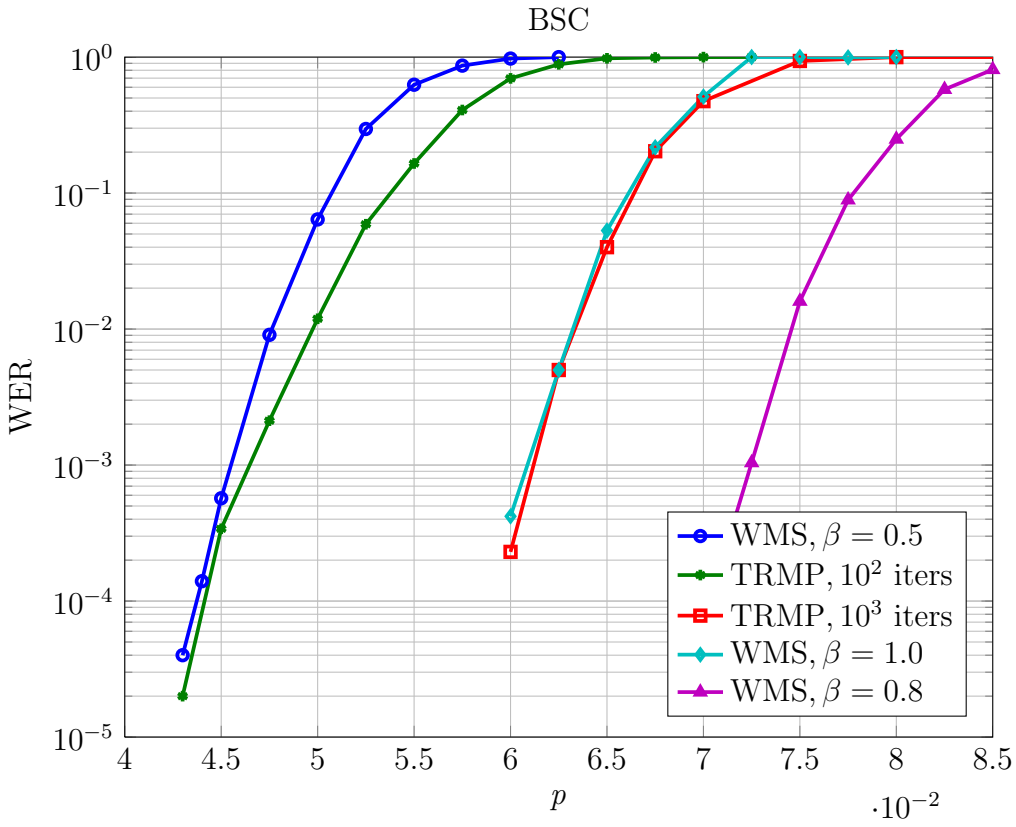

Figure 3.3: WER performance comparisons for a $(3, 6)$-regular LDPC code over the BSC.

3.6 Conclusions and Future Work

For $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes, both the AttMP algorithm and the WMS algorithm are studied. By slightly modifying the objective function of the original AttMP problem in (3.2) to an equivalent problem in (3.9), we show that the AttMP messages will converge to a fixed point when $\beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$. Further, a set of sufficient conditions (AttMP-consistency) for testing the optimality of the AttMP solutions is proposed. With the modified AttMP problem in (3.9), we show the LP and ML optimality of the AttMP solution by a simple proof if $\beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$ and the fixed point is AttMP-consistent

Similarly, when the weight factor satisfies $\beta < \frac{1}{d_v - 1}$, we show that the WMS algorithm converges to a unique fixed point. We also introduce the sufficient conditions (WMS-consistency) for the hard decisions of the WMS algorithm to be a valid codeword. By employing the construction of a dual feasible point of *Problem-P* in [66], we show that if the weight factor satisfies $\beta < \frac{1}{d_v - 1}$ and the WMS algorithm converges to a WMS-consistent fixed point, we can simplify the construction by using the converged messages. Also, we show that the dual feasible point obtained by the converged messages is a dual optimal point of *Problem-P*, and the corresponding hard decisions are the LP optimum as well as the ML decoding solution. Based on the analysis of the WMS algorithm with $\beta < \frac{1}{d_v - 1}$, the optimality of the WMS algorithm with $\beta = \frac{1}{d_v - 1}$ is also discussed. When the WMS messages with $\beta = \frac{1}{d_v - 1}$ are divergent and consistent, we show that the hard decisions are ML optimum as well. This result can be seen as the natural completion of the work initiated by Frey and Koetter in [58]. Also, our results have interesting connections with the results of [1] because their best LP thresholds also occur when $\beta = \frac{1}{d_v - 1}$ according to DE-type analysis. For weight factors $\beta > \frac{1}{d_v - 1}$, we provide examples which show that the WMS algorithm does not always return a ML codeword. In particular, the messages in Example 35 are divergent and consistent, but the hard decisions do not return a ML codeword.

In regards to future work, the most interesting open question is whether connections between LP decoding and WMS decoding can be extended beyond $\beta = \frac{1}{d_v - 1}$. In [70], Chen and Fossorier studied the optimal attenuation factor for the WMS algorithm. For example, the best β for the $(3, 6)$-regular LDPC code on the BSC is $\beta = 0.8$, and the corresponding threshold is $p = 0.083$. The DE of the WMS algorithm also shows that any extension of β beyond $\beta = \frac{1}{d_v - 1}$ will immediately provide an improved lower bound on the LP threshold. When the WMS algorithm with a

general weighting strategy for irregular LDPC codes is considered, let $d_{\mathrm{v},i}$ and β_i be the degree of the i-th bit and the weight factor for the i-th bit, respectively. By a simple extension of Theorem 46, it can be shown that the WMS algorithm for irregular LDPC codes converges to a fixed point if the weight factors satisfy $\beta_i < \frac{1}{d_{\mathrm{v},i}-1}$ for all $i \in \mathcal{V}_{\mathrm{v}}$. However, the construction of the dual optimal point of *Problem-P* using the unique fixed point of the WMS algorithm breaks down when the weight factors are not all equal. A general weighting strategy and the corresponding construction of the dual optimal point of *Problem-P* for irregular LDPC codes remains an open problem. Since suitably designed irregular LDPC codes are capacity-approaching [29], we suspect that irregular LDPC codes with a general weighting scheme might improve current estimates of the noise threshold for the LP decoding of rate-$\frac{1}{2}$ LDPC codes.

4. CONCLUDING REMARKS AND FUTURE WORK

In this section, we summarize the main contributions of this dissertation, and point out the potential future works.

4.1 Capacity Approaching GLDPC Codes Using Hard-Decision Decoding

The iterative HDD of GLDPC ensembles, based on on t-error correcting block codes, is analyzed with and without spatial coupling. In particular, we consider binary primitive BCH codes and their even-weight subcodes as component codes of GLDPC codes. Using DE analysis, noise thresholds are computed for a variety of component codes and decoding assumptions. It is proven that iterative HDD for the spatially-coupled GLDPC ensemble with BCH component codes can approach the capacity of the BSC in the high-rate regime. Finally, numerical results are presented. These results both verify the theoretical results and demonstrate the effectiveness of these codes for high-speed communication systems.

In regard to the future work, the error floor analysis of the proposed code in the finite block length case can give us a better understanding of the performance of the code. It is important for the practical design to know the error floor performance of the spatially-coupled GLDPC codes after we show good asymptotic performance. In the application of the optical communications, the system are required to have a very small bit-error probability, *e.g.*, 10^{-15}. To achieve this requirement, the designed code should not have error floor greater than the desired error probability. Although we have shown the capacity of the BSC can be approached by the proposed codes and iterative HDD algorithms, it is still unclear to us how to approach the capacity of other channels, such as bursty channel or channel with memory, using HDD algorithms. For different channels, one may need to use different component

codes to correct errors. Exploring different component codes for GLDPC codes for different channel is an open problem to us.

4.2 Convergence of Weighted Min-Sum

For $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC codes, the AttMP algorithm and the WMS algorithm, are studied in the second part of this dissertation. We first study the convergence of both algorithms. By slightly modifying the objective function of the original AttMP problem in (3.2) to an equivalent problem in (3.9), we show that the AttMP messages will converge to a fixed point when $\beta < \frac{1}{(d_{\mathrm{v}}-1)(d_{\mathrm{c}}-1)}$. Similarly, when the weight factor $\beta < \frac{1}{d_{\mathrm{v}}-1}$, we show that the WMS algorithm converges to a unique fixed point. AttMP-consistency and WMS-consistency are proposed to test the optimality of the AttMP decoding output and the WMS decoding output, respectively. If the messages passed on the edges converge to a fixed point and the fixed point satisfies the proposed sufficient condition, we show that the hard-decisions based on the fixed point are ML optimum. Based on the analysis of the WMS algorithm with $\beta < \frac{1}{d_{\mathrm{v}}-1}$, the optimality of the WMS algorithm with $\beta = \frac{1}{d_{\mathrm{v}}-1}$ is also discussed. Similar to the case of $\beta < \frac{1}{d_{\mathrm{v}}-1}$, we are able to show that the hard decisions are ML optimum as well when the messages are divergent and consistent. This result can be seen as the natural completion of the work initiated by Frey and Koetter in [58]. Also, our results have interesting connections with the results of [1] because their best LP thresholds also occur when $\beta = \frac{1}{d_{\mathrm{v}}-1}$ according to DE-type analysis.

In regards to future work, the most interesting open question is whether connections between LP decoding and WMS decoding can be extended beyond $\beta = \frac{1}{d_{\mathrm{v}}-1}$. In [70], Chen and Fossorier studied the optimal attenuation factor for the WMS algorithm. For example, the best β for the $(3, 6)$-regular LDPC code on the BSC is $\beta = 0.8$, and the corresponding threshold is $p = 0.083$. The DE of the WMS algo-

rithm also shows that any extension of β beyond $\beta = \frac{1}{d_{\mathrm{v}}-1}$ will immediately provide an improved lower bound on the LP threshold. A general weighting strategy for irregular LDPC codes is another interesting extension of this work. Let $d_{\mathrm{v},i}$ and β_i be the degree of the i-th bit and the weight factor for the i-th bit, respectively. By a simple extension of Theorem 46, it can be shown that the WMS algorithm for irregular LDPC codes converges to a fixed point if the weight factors satisfy $\beta_i < \frac{1}{d_{\mathrm{v},i}-1}$ for all $i \in \mathcal{V}_{\mathrm{v}}$. However, the construction of the dual optimal point of *Problem-P* using the unique fixed point of the WMS algorithm breaks down when the weight factors are not all equal. A general weighting strategy and the corresponding construction of the dual optimal point of *Problem-P* for irregular LDPC codes remains an open problem.

REFERENCES

[1] S. Arora, C. Daskalakis, and D. Steurer, "Message-passing algorithms and improved LP decoding," in *Proc. 41st Ann. ACM Symp. Theory of Comput. (STOC'09)*, Bethesda, MD, USA, 2009, pp. 3–12.

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

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

[4] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: The M.I.T. Press, 1963.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[6] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[7] 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.

[8] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[9] T. J. Richardson and R. 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.

[10] ——, *Modern Coding Theory*. New York, NY: Cambridge University Press, 2008.

[11] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 1, no. 9, June 1999, pp. 441–445.

[12] M. Lentmaier and K. S. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Commun. Letters*, vol. 3, pp. 248–250, Aug. 1999.

[13] N. Miladinovic and M. Fossorier, "Generalized LDPC codes and generalized stopping sets," *IEEE Trans. Commun.*, vol. 56, no. 2, pp. 201–212, 2008.

[14] *Forward Error Correction for High Bit-Rate DWDM Submarine Systems*, ITU-T Recommedation G.975.1, 2004.

[15] S. Hirst and B. Honary, "Decoding of generalised low-density parity-check codes using weighted bit-flip voting," *IEE Proc.-Commun.*, vol. 149, no. 1, pp. 1–5, 2002.

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

[17] A. Sridharan, M. Lentmaier, D. J. Costello, and K. S. Zigangirov, "Convergence analysis of a class of LDPC convolutional codes for the erasure channel," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, 2004, pp. 953–962.

[18] M. Lentmaier, A. Sridharan, K. S. Zigangirov, and D. J. Costello, "Terminated LDPC convolutional codes with thresholds close to capacity," in *Proc. IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, 2005, pp. 1372–1376.

[19] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions," in *Proc. Int. Symp. on Turbo Codes & Iterative Inform. Proc.*, 2012, pp. 51–55, arxiv preprint arXiv:1204.5703, 2012.

[20] ——, "A simple proof of threshold saturation for coupled vector recursions," in *Proc. IEEE Inform. Theory Workshop*, 2012, pp. 25–29, arxiv preprint arXiv:1208.4080.

[21] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "A proof of threshold saturation for spatially-coupled LDPC codes on BMS channels," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, Oct. 2012, pp. 176–184.

[22] A. Yedla, "Universality for multi-terminal problems via spatial coupling," Ph.D. dissertation, Texas A&M University, College Station, TX, 2012.

[23] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, Oct. 2010, pp. 347–353.

[24] D. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," Dec. 2011, arxiv preprint arXiv:1112.0708.

[25] M. I. Jordan, Ed., *Learning in graphical models*. Cambridge, MA: MIT Press, 1998.

[26] M. Mezard and A. Montanari, *Information, Physics, and Computation.* New York, NY: Oxford University Press, 2009.

[27] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 763–744, Feb. 2001.

[28] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2282–2312, 2005.

[29] T. J. Richardson, M. A. Shokrollahi, and R. 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.

[30] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. J. Viterbi, "CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Commun. Magazine*, vol. 38, no. 7, pp. 70–77, July 2000.

[31] C. Douillard, M. Jezequel, C. Berrou, N. Brengarth, J. Tousch, and N. Pham, "The turbo code standard for DVB-RCS," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, 2000, pp. 535–538.

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

[33] I. B. Djordjevic, B. Vasic, M. Ivkovic, and I. Gabitov, "Achievable information rates for high-speed long-haul optical transmission," *J. Lightwave Technol.*, vol. 23, no. 11, p. 3755, Nov. 2005.

[34] N. Abramson, "Cascade decoding of cyclic product codes," *IEEE Trans. Commun. Tech.*, vol. 16, no. 3, pp. 398–402, 1968.

[35] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP "filters" for the decoding of product and concatenated codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 3, 1993, pp. 1740–1745.

[36] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[37] M. Schwartz, P. H. Siegel, and A. Vardy, "On the asymptotic performance of iterative decoders for product codes," in *Proc. IEEE Int. Symp. Inform. Theory*, 2005, pp. 1758–1762.

[38] J. Justesen and T. Hoholdt, "Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes," in *Proc. IEEE Inform. Theory Workshop*, Sept. 2007, pp. 174–177.

[39] B. Pittel, J. Spencer, and N. Wormald, "Sudden emergence of a giant k-core in a random graph," *Journal of Combinatorial Theory-Series B*, vol. 67, no. 1, 1996.

[40] S. Janson and M. J. Luczak, "A simple solution to the k-core problem," *Random Struct. Alg*, vol. 30, pp. 50–62, 2005.

[41] A. Barg and A. Mazumdar, "On the number of errors correctable with codes on graphs," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 910–919, 2011.

[42] D. Truhachev, M. Lentmaier, and K. Zigangirov, "On braided block codes," in *Proc. IEEE Int. Symp. Inform. Theory*, 2003, p. 32.

[43] A. J. Feltstrom, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, "Braided block codes," *IEEE Trans. Inform. Theory*, vol. 55, no. 6, pp. 2640–2658, 2009.

[44] J. Justesen, K. J. Larsen, and L. A. Pedersen, "Error correcting coding for OTN," *IEEE Commun. Magazine*, vol. 48, no. 9, pp. 70–75, 2010.

[45] J. Justesen, "Performance of product codes and related structures with iterated decoding," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, 2011.

[46] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase Codes: FEC for 100 Gb/s OTN," *J. Lightwave Technol.*, vol. 30, no. 1, pp. 110–117, 2012.

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

[48] M. G. Kim and J. H. Lee, "Undetected error probabilities of binary primitive BCH codes for both error correction and detection," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 575–580, 1996.

[49] V. M. Sidel'nikov, "Weight spectrum of binary Bose-Chaudhuri-Hoquinghem codes," *Problems of Inform. Transm.*, vol. 7, no. 1, pp. 14–22, 1971.

[50] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.

[51] S. H. Hassani, N. Macris, and R. Urbanke, "Threshold saturation in spatially coupled constraint satisfaction problems," 2011, arxiv preprint arXiv:1112.6320.

[52] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, Eds., *NIST Handbook of Mathematical Functions*. New York, NY: Cambridge University Press, 2010, [Online]. Available: http://dlmf.nist.gov/.

[53] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *Proc. IEEE Int. Symp. Inform. Theory*, 2012, pp. 2696–2700.

[54] H.-K. Hwang and V. Zacharovas, "Uniform asymptotics of Poisson approximation to the Poisson-binomial distribution," *Theory of Probab. & its Appl.*, vol. 55, no. 2, pp. 198–224, 2011.

[55] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University, Department of Electrical Engineering, S-581 83 Linköping, Sweden, 1996.

[56] N. Wiberg, H.-A. Loeliger, and R. Kötter, "Codes and iterative decoding on general graphs," *Eur. Trans. Telecom.*, vol. 6, no. 5, pp. 513–525, Sept. – Oct. 1995.

[57] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms.* New York, NY: Cambridge University Press, 2003.

[58] B. J. Frey and R. Koetter, "Exact inference using the attenuated max-product algorithm," in *Advanced Mean Field Methods: Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2000.

[59] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on trees: Message-passing and linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[60] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954–972, March 2005.

[61] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. Int. Symp. on Turbo Codes & Iterative Inform. Proc.*, Brest, France, Sept. 2003, pp. 75–82.

[62] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Int. Symp. Inform. Theory and its Appl.*, Parma, Italy, Oct. 2004.

[63] ——, "Bounds on the threshold of linear programming decoding," in *Proc. IEEE Inform. Theory Workshop*, Punta del Este, Uruguay, March 2006, pp. 175–179.

[64] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inform. Theory*, vol. 53, no. 1, pp. 82–89, Jan. 2007.

[65] C. Daskalakis, A. G. Dimakis, R. M. Karp, and M. J. Wainwright, "Probabilistic analysis of linear programming decoding," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3565–3578, 2008.

[66] R. Koetter and P. O. Vontobel, "On the block error probability of LP decoding of LDPC codes," in *Proc. 1st Annual Workshop on Inform. Theory and its Appl.*, San Diego, CA, Feb. 2006.

[67] N. Halabi and G. Even, "LP decoding of regular LDPC codes in memoryless channels," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, Feb. 2011.

[68] ——, "Linear-programming decoding of Tanner codes with local-optimality certificates," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, MA, July 2012, pp. 2696–2700.

[69] ——, "Hierarchies of local-optimality characterizations in decoding Tanner codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, MA, July 2012, pp. 2701–2705.

[70] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Letters*, vol. 6, no. 5, pp.

208–210, 2002.

[71] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4422–4437, Dec. 2007.

[72] N. L. Carothers, *Real Analysis*. New York, NY: Cambridge University Press, 1999.

[73] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley and Sons, 1994.

[74] Y.-Y. Jian and H. D. Pfister, "Convergence of weighted min-sum decoding via dynamic programming on coupled trees," in *Proc. Int. Symp. on Turbo Codes & Iterative Inform. Proc.*, Brest, France, Sept. 2010.

[75] D. Blackwell, "Discrete dynamic programming," *Ann. Math. Stats.*, vol. 33, pp. 719–726, 1962.

[76] V. Kolmogorov and M. J. Wainwright, "On the optimality of tree-reweighted max-product message-passing," in *Proc. Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, UK, July 2005.

[77] P. O. Vontobel, "A factor-graph-based random walk, and its relevance for LP decoding analysis and Bethe entropy characterization," in *Proc. Annual Workshop on Inform. Theory and its Appl.*, San Diego, CA, Feb. 2010.

APPENDIX A

PROOFS OF LEMMAS

A.1 Proof of Lemma 9

Proof. Let

$$F(n,i,\delta,j) \triangleq \frac{n-l(i,\delta,j)}{n} 2^{-mt} \binom{n}{l(i,\delta,j)} \times \binom{l(i,\delta,j)}{l(i,\delta,j)-j} \binom{n-l(i,\delta,j)-1}{\delta-1-j} \binom{n-1}{i}^{-1}.$$

By using (2.17), for a fixed $i \geq t$, one can rewrite the last term of (2.13) as

$$\left(1+O\left(n^{-0.1}\right)\right) \sum_{\delta=1}^{t} \sum_{j=0}^{\delta-1} F(n,i,\delta,j),$$

and $P_n(i)$ is

$$P_n(i) = 1 - \left(1+O\left(n^{-0.1}\right)\right) \sum_{\delta=1}^{t} \sum_{j=0}^{\delta-1} F(n,i,\delta,j). \tag{A.1}$$

By the fact that $2^m = n+1$, it is simple to show that

$$\begin{aligned} F(n,i,\delta,j) &= \frac{n-l(i,\delta,j)}{n} \frac{1}{(n+1)^t} \frac{n!}{(n-l(i,\delta,j))!l(i,\delta,j)!} \\ &\quad \times \frac{l(i,\delta,j)!}{(l(i,\delta,j)-j)!j!} \frac{(n-l(i,\delta,j)-1)!}{(n-l(i,\delta,j)-\delta+j)!(\delta-1-j)!} \frac{i!(n-1-i)!}{(n-1)!} \\ &= \frac{1}{(n+1)^t} \frac{i!}{(l(i,\delta,j)-j)!} \frac{(n-1-i)!}{(n-l(i,\delta,j)-\delta+j)!} \frac{1}{j!(\delta-1-j)!} \qquad \text{(A.2)} \\ &= \frac{1}{(n+1)^t} \frac{i!}{(i-\delta+j+1)!} \frac{(n-1-i)!}{(n-i-j-1)!} \frac{1}{j!(\delta-1-j)!}, \qquad \text{(A.3)} \end{aligned}$$

where (A.3) is obtained by substituting (2.12) into (A.2). When $j < \delta - 1$, (A.3) can be written as

$$\frac{1}{j!(\delta-1-j)!}\frac{1}{(n+1)^{t-\delta+1}}\left(\prod_{k=0}^{\delta-j-2}\frac{i-k}{n+1}\right)\left(\prod_{k'=0}^{j-1}\frac{n-1-i-k'}{n+1}\right). \tag{A.4}$$

On the other hand, when $j = \delta - 1$, (A.3) becomes

$$\frac{1}{(n+1)^{t-\delta+1}}\frac{1}{(\delta-1)!}\left(\prod_{k'=0}^{\delta-2}\frac{n-1-i-k'}{n+1}\right). \tag{A.5}$$

Substituting (A.4) and (A.5) into (A.1), we have

$$\begin{aligned}
P_n(i) &= 1-\left(1+O\left(n^{-0.1}\right)\right)\left(\sum_{j=0}^{t-1}F(n,i,t,j)+\sum_{\delta=1}^{t-1}\sum_{j=0}^{\delta-1}F(n,i,\delta,j)\right)\\
&= 1-\left(1+O\left(n^{-0.1}\right)\right)\frac{1}{n+1}\left(\frac{1}{(t-1)!}\prod_{k'=0}^{t-2}\frac{n-1-i-k'}{n+1}\right.\\
&\qquad\left.+\sum_{j=0}^{t-2}\frac{1}{j!(t-1-j)!}\left(\prod_{k=0}^{t-j-2}\frac{i-k}{n+1}\right)\left(\prod_{k'=0}^{j-1}\frac{n-1-i-k'}{n+1}\right)\right)+O\left(n^{-2}\right)\\
&> 1-\left(1+O\left(n^{-0.1}\right)\right)n^{-1}\left(\sum_{j=0}^{t-1}\frac{1}{j!(t-j-1)!}\right)+O\left(n^{-2}\right).
\end{aligned}$$

By the fact that $\sum_{j=0}^{t-1}\frac{1}{j!(t-j-1)!} \le 2$, we have $\lim_{n\to\infty} P_n(i) = 1$.

For the analysis of $nQ_n(i)$, we also define

$$\begin{aligned}
K(n,i,\delta,j) \triangleq{}& \frac{l(i,\delta,j)-1}{n}2^{-mt}\binom{n}{l(i,\delta,j)-1}\\
&\times\binom{l(i,\delta,j)-2}{l(i,\delta,j)-j-1}\binom{n-l(i,\delta,j)+1}{\delta-1}\binom{n-1}{i}^{-1}.
\end{aligned} \tag{A.6}$$

Then, we have

$$nQ_n(i) = \sum_{\delta=1}^{t} \sum_{j=0}^{\delta} \left(1 + O\left(n^{-0.1}\right)\right) nK(n, i, \delta, j). \tag{A.7}$$

We first show that $nK(n, i, \delta, j)$ is bounded. By the similar simplification as (A.3), one can have

$$nK(n, i, \delta, j) = \frac{n}{(n+1)^t} \frac{i!}{(i-\delta+j)!} \frac{(n-1-i)!}{(n-i-j)!} \frac{1}{(j-1)!(\delta-j)!}. \tag{A.8}$$

When $j < \delta$, the RHS of (A.8) can be simplified by

$$\frac{1}{(j-1)!(\delta-j)!} \frac{n}{(n+1)^{t-\delta+1}} \left(\prod_{k=0}^{\delta-j-1} \frac{i-k}{n+1} \right) \left(\prod_{k'=0}^{j-2} \frac{n-1-i-k'}{n+1} \right). \tag{A.9}$$

On the other hand, when $j = \delta$, (A.8) becomes

$$\frac{1}{(\delta-1)!} \frac{n}{(n+1)^{t-\delta+1}} \left(\prod_{k'=0}^{\delta-2} \frac{n-1-i-k'}{n+1} \right). \tag{A.10}$$

It is obvious that both (A.9) and (A.10) are upper bounded by 1, and therefore, $nQ_n(i)$ is bounded for all $t+1 \le i \le n-t-1$.

To show (2.20), we first introduce some upper bounds for $nK(n, i, \delta, j)$. When j and δ satisfy $\delta < t$ and $0 \le j \le \delta$, both (A.9) and (A.10) imply that $nK(n, i, \delta, j) < \frac{n}{n^{t-\delta+1}} \le \frac{1}{n}$ for all $t+1 \le i \le n-t-1$. Also, by substituting $j = \delta = t$ into (A.10), we have $nK(n, i, t, t) < \frac{1}{(t-1)!}$ for all $t+1 \le i \le n-t-1$. When $\delta = t$ but $j < \delta$, the interval $t+1 \le i \le \lfloor \sqrt{n} \rfloor$ is considered. From (A.9), we know

$$nK(n, i, t, j) \le \frac{\lfloor \sqrt{n} \rfloor}{n+1} < \frac{1}{\sqrt{n}}.$$

From (A.7), $nQ_n(i)$ for $t+1 \le i \le \lfloor \sqrt{n} \rfloor$ can be upper bounded by

$$\begin{aligned} nQ_n(i) &= \sum_{\delta=1}^{t} \sum_{j=0}^{\delta} \left(1 + O\left(n^{-0.1}\right)\right) nK(n,i,\delta,j) \\ &\le \left(1 + O\left(n^{-0.1}\right)\right) \left(\sum_{\delta=1}^{t-1} \sum_{j=0}^{\delta} \frac{1}{n} + \sum_{j=0}^{t-1} \frac{1}{\sqrt{n}} + \frac{1}{(t-1)!} \right) \\ &= \frac{1}{(t-1)!} \left(1 + O\left(n^{-0.1}\right)\right). \end{aligned} \tag{A.11}$$

From (A.11) and the fact that $nQ_n(i) \ge 0$, we know $\lim_{n\to\infty} nQ_n(i) = 0$.

A.2 Proof of Lemma 10

Proof. Since $P_n(i) = 0$ for $0 \le i \le t-1$ and $P_n(i) < 1$ for all $i \ge t$, we know that

$$1 - E\left[P_n(X_n)\right] \ge \sum_{i=0}^{t-1} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i},$$

and $\lim_{n\to\infty} E\left[P_n(X_n)\right] \le \phi(\lambda; t-1)$. With the convention that $\binom{n}{k} = 0$ if $k > n$, we know that, for any fixed $T \gg 1$,

$$E\left[P_n(X_n)\right] \ge \sum_{i=t}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i} P_n(i).$$

Then

$$\begin{aligned} \lim_{n\to\infty} E\left[P_n(X_n)\right] &\ge \lim_{n\to\infty} \sum_{i=t}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i} P_n(i) \\ &= \phi(\lambda; t-1) - \phi(\lambda; T). \end{aligned}$$

Since T is arbitrary and by Markov's inequality $\phi(\lambda;T) \leq \frac{\lambda}{T+1}$, we have

$$\lim_{n\to\infty} E\left[P_n(X_n)\right] = \phi(\lambda; t-1).$$

By the fact that $Q_n(i) \leq 1$ for all $i \geq t+1$, one can show that, for any $T \gg 1$,

$$\begin{aligned} E\left[Q_n(X_n)\right] \leq & \sum_{i=t+1}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} Q_n(i) \\ & + \sum_{i=T+1}^{\infty} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i}. \end{aligned}$$

From Lemma 9, we have $\lim_{n\to\infty} E\left[Q_n(X_n)\right] \leq \phi(\lambda;T)$. Since $Q_n(i) \geq 0$ for all i, we have $\lim_{n\to\infty} E\left[Q_n(X_n)\right] = 0$.

A.3 Proof of Lemma 11

Proof. By using the convention that $\binom{n}{k} = 0$ when ever $k > n$, for any $T \gg 1$, we can rewrite $E\left[nQ_n(X_n)\right]$ as

$$\begin{aligned} E\left[nQ_n(X_n)\right] = & \sum_{i=t+1}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} nQ_n(i) \\ & + \sum_{i=T+1}^{\infty} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} nQ_n(i). \end{aligned}$$

Then, by the Poisson theorem [50, pp. 113] and Lemma 9 we know

$$\begin{aligned} \lim_{n\to\infty} E\left[nQ_n(X_n)\right] &\geq \lim_{n\to\infty} \sum_{i=t+1}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} nQ_n(i) \\ &= \frac{1}{(t-1)!}\left(\phi(\lambda;t) - \phi(\lambda;T)\right). \end{aligned}$$

Since $nQ_n(i)$ is bounded for $t+1 \leq i \leq n-2-t$ according to Lemma 9, there exists a constant $0 < C < \infty$ independent of n such that $nQ_n(i) \leq C$ for $t+1 \leq i \leq n-2-t$. Also, we know $nQ_n(i) = 0$ for $0 \leq i \leq t$ and $nQ_n(i) = n$ for $n-t-1 \leq i \leq n-1$. Thus, $E\left[nQ_n(X_n)\right]$ can be upper bounded by

$$\begin{aligned} E\left[nQ_n(X_n)\right] \leq & \sum_{i=t+1}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} nQ_n(i) \\ & + C \sum_{i=T+1}^{\infty} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} \\ & + n \sum_{i=n-t-1}^{n-1} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i}. \end{aligned} \tag{A.12}$$

By the Chernoff bound of the binomial distribution $B(n-1, \frac{\lambda_n}{n-1})$, the last term of (A.12) can be upper bounded by

$$n \sum_{i=n-t-1}^{n-1} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} \leq n \left(\frac{e\lambda_n}{n-1-t}\right)^{n-t-1} e^{-\lambda_n}. \tag{A.13}$$

Since $\lambda_n \to \lambda$ and $\lambda < \infty$, we know that there exists a $N_0 > 0$ and a $\epsilon > 0$ such that $\lambda_n \leq \lambda + \epsilon$ whenever $n > N_0$. Thus, we have

$$0 \leq \lim_{n\to\infty} n \left(\frac{e\lambda_n}{n-1-t}\right)^{n-t-1} e^{-\lambda_n} \leq e^{-\lambda} \lim_{n\to\infty} n \left(\frac{e(\lambda+\epsilon)}{n-1-t}\right)^{n-t-1} = 0,$$

and

$$\begin{aligned}\lim_{n\to\infty} E\left[nQ_n(X_n)\right] &\le \lim_{n\to\infty} \sum_{i=t+1}^{T} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} nQ_n(i) \\ &\quad + C \lim_{n\to\infty} \sum_{i=T+1}^{\infty} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} \\ &= \frac{1}{(t-1)!}\phi(\lambda;t) + \left(C - \frac{1}{(t-1)!}\right)\phi(\lambda;T).\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\frac{1}{(t-1)!}\phi(\lambda;t) - \frac{1}{(t-1)!}\phi(\lambda;T) \\ &\qquad \le \lim_{n\to\infty} E\left[nQ_n(X_n)\right] \le \frac{1}{(t-1)!}\phi(\lambda;t) + \left(C - \frac{1}{(t-1)!}\right)\phi(\lambda;T).\end{aligned}$$

Since C is independent of n and T is arbitrary, we know $\lim_{n\to\infty} E\left[nQ_n(X_n)\right] = \frac{1}{(t-1)!}\phi(\lambda;t)$.

A.4 Proof of Lemma 14

Proof. To show the lemma, it suffices to show that $\lim_{n\to\infty} \tilde{P}_n(i) = 1$ for $i \ge t$, $\lim_{n\to\infty} E[\tilde{Q}_n(X_n)] = 0$, and

$$\lim_{n\to\infty} E\left[n\tilde{Q}_n(X_n)\right] = \begin{cases} \frac{1}{(t-1)!}\psi\left(\lambda;t\right) & \text{if } t \text{ is even,} \\ \frac{1}{(t-1)!}\varphi\left(\lambda;t\right) & \text{if } t \text{ is odd.} \end{cases} \tag{A.14}$$

Let $\mathcal{L}_1(i,\delta) \triangleq \{j \in [0,\delta] \mid l(i,\delta,j) = 0 \bmod 2\}$ be the set of all j such that $l(i,\delta,j)$ is an even number. Since $A_\ell = 0$ for all odd ℓ, we have

$$\tilde{P}_n(i) = 1 - \sum_{\delta=1}^{t} \sum_{j\in\mathcal{L}_1(i,\delta)} \frac{n-l(i,\delta,j)}{n} A_{l(i,\delta,j)}\Theta(n,i,\delta,j) \ge P_n(i).$$

Likewise, we define $\mathcal{L}_2(i,\delta) \triangleq \{j \in [0,\delta] \mid (l(i,\delta,j)-1) = 0 \bmod 2\}$ be the set of all j such that $l(i,\delta,j)-1$ is an even number. Then,

$$\tilde{Q}_n(i) = \sum_{\delta=1}^{t} \sum_{j \in \mathcal{L}_2(i,\delta)} \frac{l(i,\delta,j)-1}{n} A_{l(i,\delta,j)-1} \Lambda(n,i,\delta,j) \leq Q_n(i).$$

From Lemma 9 and Lemma 10, we immediately have $\lim_{n\to\infty} \tilde{P}_n(i) = 1$ for $i \geq t$, and $\lim_{n\to\infty} E[\tilde{Q}_n(X_n)] = 0$.

From (A.6) and (A.7, we have

$$n\tilde{Q}_n(i) = \sum_{\delta=1}^{t} \sum_{j \in \mathcal{L}_2(i,\delta)} \left(1 + O\left(n^{-0.1}\right)\right) nK(n,i,\delta,j),$$

When t is even, and i is odd, one can show that $\mathcal{L}_2(i,t) = \emptyset$. From (A.10), we know

$$n\tilde{Q}_n(i) \leq \sum_{\delta=1}^{t-1} \sum_{j \in 0}^{\delta} \left(1 + O\left(n^{-0.1}\right)\right) nK(n,i,\delta,j) = O\left(n^{-1}\right).$$

When both t and i are even numbers, one can show $t \in \mathcal{L}_2(i,t)$. By the same argument in the proof of Lemma 9, we know $n\tilde{Q}_n(i)$ is upper bounded by a constant for all even i with $i \geq t+1$, and for all even $t+1 \leq i \leq \sqrt{n}$, $n\tilde{Q}_n(i) = \frac{1}{(t-1)!}(1 + O(n^{-0.1}))$. Let $\mathbb{N}_e$ be the set of even natural numbers, and $\mathbb{N}_o$ be the set of odd nature numbers. Then, when t is even, we have

$$\begin{aligned} E\left[n\tilde{Q}_n(X_n)\right] &= \sum_{i \in \mathbb{N}_e, i \geq t+2} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) \\ &\quad + \sum_{i \in \mathbb{N}_o, i \geq t+1} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) \\ &= \sum_{i \in \mathbb{N}_e, i \geq t+2} \binom{n-1}{i} \left(\frac{\lambda_n}{n-1}\right)^i \left(1 - \frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) + O\left(\frac{1}{n}\right). \end{aligned}$$

By the same calculation in the proof of Lemma 11, we have

$$\begin{aligned}\lim_{n\to\infty} E\left[n\tilde{Q}_n(X_n)\right] &= \frac{1}{(t-1)!}\sum_{i\in\mathbb{N}_e, i\geq t+2}\frac{\lambda^i e^{-\lambda}}{i!} \\ &= \frac{1}{(t-1)!}\left(\sum_{i\in\mathbb{N}_e, i\geq t+2}\frac{\lambda^i e^{-\lambda}}{i!} - \sum_{i'=0}^{\frac{t}{2}}\frac{\lambda^{2i}e^{-\lambda}}{(2i)!}\right) \\ &\overset{\text{(a)}}{=} \frac{1}{(t-1)!}\left(\frac{\left(1-e^{-2\lambda}\right)}{2} - \sum_{i=0}^{\frac{t}{2}}\frac{\lambda^{2i}e^{-\lambda}}{(2i)!}\right),\end{aligned}$$

where (a) follows from the fact that $\sum_{i\in\mathbb{N}_e, i\geq t+2}\frac{\lambda^i e^{-\lambda}}{i!} = \frac{1}{2}(1-e^{-2\lambda})$. Thus, we showed (A.14) when t is a even number.

When t is a odd number, we know $\mathcal{L}_2(i,t) = \emptyset$ for all even i. Then, one can have $n\tilde{Q}_n(i) < \frac{1}{n}$ for all even $i \geq t+1$, and $n\tilde{Q}_n(i) = \frac{1}{(t-1)!}(1+O(n^{-0.1}))$ for all odd $t+1 \leq i \leq \sqrt{n}$. With the same argument in showing the case where t is even, we have

$$\begin{aligned}E\left[n\tilde{Q}_n(X_n)\right] &= \sum_{i\in\mathbb{N}_o, i\geq t+2}\binom{n-1}{i}\left(\frac{\lambda_n}{n-1}\right)^i\left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) \\ &\quad + \sum_{i\in\mathbb{N}_e, i\geq t+1}\binom{n-1}{i}\left(\frac{\lambda_n}{n-1}\right)^i\left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) \\ &= \sum_{i\in\mathbb{N}_o, i\geq t+2}\binom{n-1}{i}\left(\frac{\lambda_n}{n-1}\right)^i\left(1-\frac{\lambda_n}{n-1}\right)^{n-1-i} n\tilde{Q}_n(i) + O\left(\frac{1}{n}\right).\end{aligned}$$

Then,

$$\begin{aligned}\lim_{n\to\infty} E\left[n\tilde{Q}_n(X_n)\right] &= \frac{1}{(t-1)!}\sum_{i\in\mathbb{N}_o, i\geq t+2}\frac{\lambda^i e^{-\lambda}}{i!} \\ &= \frac{1}{(t-1)!}\left(\frac{\left(1+e^{-2\lambda}\right)}{2} - \sum_{i=0}^{\frac{t-1}{2}}\frac{\lambda^{2i+1}e^{-\lambda}}{(2i+1)!}\right).\end{aligned}$$

This shows the other case of (A.14), and also, complete the proof the lemma.

A.5 Proof of Lemma 20

Proof. Since $V_n(x)$ is a continuous function in $(0, 1]$, we prove that $\hat{x}_n^{**}$ exists by showing that $V_n\left(\frac{2t}{n}\right) < 0$ and $V_n\left(\frac{2t-2}{n}\right) > 0$ when $n \geq n_0$ and $t \geq t_0$. From (2.30), one can readily have

$$V_n\left(\frac{2t}{n}\right) = -\frac{2}{nB(t, n-t)}\left(\frac{2t}{n}\right)^t\left(1-\frac{2t}{n}\right)^{n-t} < 0. \tag{A.15}$$

We simplify $V_n\left(\frac{2t-2}{n}\right)$ by

$$\begin{aligned}
&V_n\left(\frac{2t-2}{n}\right) \\
&\quad= \frac{2}{n} I_{\frac{2t-2}{n}}(t, n-t) - \frac{2}{nB(t, n-t)}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t} \\
&\quad\overset{\text{(a)}}{=} \frac{2}{n}\left(I_{\frac{2t-2}{n}}(t, n-t) - t\left(1-\frac{2t-2}{n}\right)\binom{n-1}{t}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t-1}\right) \\
&\quad\geq \frac{2}{n}\left(I_{\frac{2t-2}{n}}(t, n-t) - t\binom{n-1}{t}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t-1}\right),
\end{aligned}$$

where (a) follows from the definition of $B(t, n-t)$ in (2.24). Since $t_0 > 3$, one can apply Chernoff bound for the lower tail of the binomial distribution by

$$1 - I_{\frac{2t-2}{n}}(t, n-t) \leq \left(2-\frac{2}{t}\right)^t e^{-(t-2)}.$$

Thus, we know

$$\begin{aligned} & V_n\left(\frac{2t-2}{n}\right) \\ & \geq \frac{2}{n}\left(1-\left(2-\frac{2}{t}\right)^t e^{-(t-2)} - t\binom{n-1}{t}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t-1}\right) \\ & = \frac{2}{n}\Psi(n;t), \end{aligned}$$

where $\Psi(n;t) \triangleq 1 - e^{-(t-2)}\left(2-\frac{2}{t}\right)^t - t\binom{n-1}{t}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t-1}$. By the Poisson theorem [50, pp. 113] and the fact that $t! \leq t^t e^{-t}$ [10, pp. 30], one can show

$$\lim_{n\to\infty}\binom{n-1}{t}\left(\frac{2t-2}{n}\right)^t\left(1-\frac{2t-2}{n}\right)^{n-t-1} = \frac{(2t-2)^t}{t!}e^{-(2t-2)} \geq \frac{(2t-2)^t}{t^t e^{-t}}e^{-(2t-2)}.$$

We first let $n \to \infty$, and then $t \to \infty$. The function $\Psi(n;t)$ can be lower bounded by

$$\begin{aligned} \lim_{t\to\infty}\lim_{n\to\infty}\Psi(n;t) &\geq \lim_{t\to\infty}\left(1-\left(2-\frac{2}{t}\right)^t e^{-(t-2)} - t\left(2-\frac{2}{t}\right)^t e^{-(t-2)}\right) \\ &= 1 - \lim_{t\to\infty}(1+t)\left(2-\frac{2}{t}\right)^t e^{-(t-2)} \\ &= 1. \end{aligned}$$

Therefore, there exists a pair (n_0, t_0) with $t_0 > 3$ and $n_0 > t_0$ such that $\Psi(n;t) > 0$ for all $n \geq n_0$ and $t \geq t_0$, and thus, $V_n\left(\frac{2t-2}{n}\right) > 0$. Since (A.15) holds for any $n > 0$ and $t > 0$, we conclude that $\hat{x}^{**}$ exists and $\frac{2t-2}{n} \leq \hat{x}_n^{**} \leq \frac{2t}{n}$.

By the fact that $\hat{p}_n^{**} = \frac{\hat{x}_n^{**}}{\hat{f}_n(\hat{x}_n^{**})}$ and $\hat{f}_n(x) \leq 1$ for all $x \in [0,1]$, we know

$$\hat{p}_n^{**} \geq \hat{x}_n^{**} \geq \frac{2t-2}{n}.$$

A.6 Proof of Lemma 26

Proof. We first rewrite $E\left[nQ_n(X_n)\right]$ by

$$\begin{aligned} E\left[nQ_n(X_n)\right] = &\sum_{i=t+1}^{\lfloor\sqrt{n}\rfloor} \binom{n-1}{i} \left(\frac{\lambda}{n-1}\right)^i \left(1-\frac{\lambda}{n-1}\right)^{n-i-1} nQ_n(i) \\ &+ \sum_{i=\lfloor\sqrt{n}\rfloor+1}^{n-t-2} \binom{n-1}{i} \left(\frac{\lambda}{n-1}\right)^i \left(1-\frac{\lambda}{n-1}\right)^{n-i-1} nQ_n(i) \\ &+ n \sum_{i=n-t-2}^{n-1} \binom{n-1}{i} \left(\frac{\lambda}{n-1}\right)^i \left(1-\frac{\lambda}{n-1}\right)^{n-i-1}. \end{aligned}$$

From Lemma 9 and (2.20), we can upper bound $E\left[nQ_n\left(X_n\right)\right]$ by

$$\begin{aligned} E\left[nQ_n(X_n)\right] \leq &\left(\frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}(t+1, n-t-1) \\ &+ C_1 I_{\frac{\lambda}{n-1}}\left(\lfloor\sqrt{n}\rfloor+1, n-\lfloor\sqrt{n}\rfloor-1\right) + nI_{\frac{\lambda}{n-1}}\left(n-t-2, t-2\right), \end{aligned} \tag{A.16}$$

where C_1 is a constant. By applying Chernoff bound, the second term of (A.16) is upper bounded by

$$\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor} e^{-\lambda}.$$

Thus, with the upper bound (A.13) for the last term of (A.16), we have

$$\begin{aligned} E\left[nQ_n(X_n)\right] \leq &\left(\frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}(t+1, n-t-1) \\ &+ C_1 \left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor} e^{-\lambda} + n\left(\frac{e\lambda}{n-1-t}\right)^{n-t-1} e^{-\lambda} \end{aligned}$$

By the fact that, for any $\lambda > 0$,

$$\lim_{n\to\infty} \frac{\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}}{I_{\frac{\lambda}{n-1}}(t+1, n-t-1)} \leq \lim_{n\to\infty} \frac{\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}}{\left(\frac{\lambda}{n-1}\right)^{t+1}\left(1-\frac{\lambda}{n-1}\right)^{n-t-2}} = 0,$$

and

$$\begin{aligned}
\lim_{n\to\infty}\lim_{\lambda\to 0} \frac{\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}}{I_{\frac{\lambda}{n-1}}(t+1, n-t-1)} &\leq \lim_{n\to\infty}\lim_{\lambda\to 0} \frac{\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}}{\left(\frac{\lambda}{n-1}\right)^{t+1}\left(1-\frac{\lambda}{n-1}\right)^{n-t-2}} \\
&= \lim_{n\to\infty}\lim_{\lambda\to 0} \frac{(n-1)^{t+1}\left(\frac{e}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}\lambda^{\lfloor\sqrt{n}\rfloor-t-1}}{\left(1-\frac{\lambda}{n-1}\right)^{n-t-2}} \\
&= \lim_{\lambda\to 0}\lim_{n\to\infty} \frac{(n-1)^{t+1}\left(\frac{e}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor}\lambda^{\lfloor\sqrt{n}\rfloor-t-1}}{\left(1-\frac{\lambda}{n-1}\right)^{n-t-2}} = 0,
\end{aligned}$$

we have $\left(\left(\frac{e\lambda}{\lfloor\sqrt{n}\rfloor}\right)^{\lfloor\sqrt{n}\rfloor} + n\left(\frac{e\lambda}{n-1-t}\right)^{n-t-1}\right) I^{-1}_{\frac{\lambda}{n-1}}(t+1, n-t-1) = O(n^{-1})$, and thus,

$$E\left[nQ_n(X_n)\right] \leq \left(\frac{1}{(t-1)!} + O\left(n^{-0.1}\right)\right) I_{\frac{\lambda}{n-1}}(t+1, n-t-1).$$

This concludes the proof of the lemma.

A.7 Proof of Lemma 45

Proof. Since $\prod_{m\in N(k)\setminus i} \operatorname{sgn}(\mu_{m\to k})\operatorname{sgn}(\nu_{m\to k})$ can be ± 1, we need to verify the inequality

$$\max_{m\in N(k)\setminus i} |\mu_{m\to k} - \nu_{m\to k}| \geq \left| \min_{m\in N(k)\setminus i} |\mu_{m\to k}| - \min_{m\in N(k)\setminus i} |\nu_{m\to k}| \right| \tag{A.17}$$

whenever an even number of signs of $\mu_{m\to k}$ and $\nu_{m\to k}$ on $m \in N(k)\setminus i$ do not match. On the other hand, if an odd number of signs on $m \in N(k)\setminus i$ do not match, we

need to show

$$\max_{m\in N(k)\backslash i} |\mu_{m\to k} - \nu_{m\to k}| \geq \left| \min_{m\in N(k)\backslash i} |\mu_{m\to k}| + \min_{m\in N(k)\backslash i} |\nu_{m\to k}| \right|. \tag{A.18}$$

To show (A.17), let m_1, m_2, and $m^\star$ be bit nodes in sets $\mathcal{M}_\mu \triangleq \arg\min_{m\in N(k)\backslash i} |\mu_{m\to k}|$, $\mathcal{M}_\nu \triangleq \arg\min_{m\in N(k)\backslash i} |\nu_{m\to k}|$, and $\mathcal{M}^\star \triangleq \arg\max_{m\in N(k)\backslash i} |\mu_{m\to k} - \nu_{m\to k}|$, respectively. Notice that

$$\begin{aligned} |\mu_{m^\star\to k} - \nu_{m^\star\to k}| &\geq \max_{m\in N(k)\backslash i} \big| |\mu_{m\to k}| - |\nu_{m\to k}| \big| \\ &= \max_{m\in N(k)\backslash i} \big| |\nu_{m\to k}| - |\mu_{m\to k}| \big|. \end{aligned}$$

Consider the case when $|\mu_{m_1\to k}| \geq |\nu_{m_2\to k}|$. Since $|\mu_{m_2\to k}| \geq |\mu_{m_1\to k}| \geq |\nu_{m_2\to k}|$, it follows that

$$\begin{aligned} |\mu_{m^\star\to k} - \nu_{m^\star\to k}| &\geq \max_{m\in N(k)\backslash i} \big| |\mu_{m\to k}| - |\nu_{m\to k}| \big| \\ &\geq \big| |\mu_{m_2\to k}| - |\nu_{m_2\to k}| \big| \\ &\geq \big| |\mu_{m_1\to k}| - |\nu_{m_2\to k}| \big|. \end{aligned}$$

On the other hand, when $|\mu_{m_1\to k}| \leq |\nu_{m_2\to k}|$, one can show $|\mu_{m_1\to k}| \leq |\nu_{m_2\to k}| \leq |\nu_{m_1\to k}|$. Therefore, we have

$$\begin{aligned} |\mu_{m^\star\to k} - \nu_{m^\star\to k}| &\geq \max_{m\in N(k)\backslash i} \big| |\mu_{m\to k}| - |\nu_{m\to k}| \big| \\ &\geq \big| |\nu_{m_1\to k}| - |\mu_{m_1\to k}| \big| \\ &\geq \big| |\nu_{m_2\to k}| - |\mu_{m_1\to k}| \big| \\ &= \big| |\mu_{m_1\to k}| - |\nu_{m_2\to k}| \big|. \end{aligned}$$

Combining these two cases implies (A.17).

To show (A.18), let $\mathcal{M}_k = \{m \in N(k) \setminus i : \mu_{m\to k}\nu_{m\to k} < 0\}$ be the set of indices such that $\mu_{m\to k}$ and $\nu_{m\to k}$ have different signs. Notice that

$$\begin{aligned}
\max_{m\in N(k)\setminus i} |\mu_{m\to k} - \nu_{m\to k}| &\geq \max_{m\in\mathcal{M}_k} |\mu_{m\to k} - \nu_{m\to k}| \\
&\geq \max_{m\in\mathcal{M}_k} \big||\mu_{m\to k}| + |\nu_{m\to k}|\big| \\
&\geq \left|\min_{m\in\mathcal{M}_k} |\mu_{m\to k}| + \min_{m\in\mathcal{M}_k} |\nu_{m\to k}|\right| \\
&\geq \left|\min_{m\in N(k)\setminus i} |\mu_{m\to k}| + \min_{m\in N(k)\setminus i} |\nu_{m\to k}|\right|.
\end{aligned}$$

This completes the proof.

A.8 Proof of Lemma 54

Proof. Let $\boldsymbol{\mu}' \triangleq \mathtt{W}_\delta[\boldsymbol{\mu}]$ and $\boldsymbol{\nu}' \triangleq \mathtt{W}_\delta[\boldsymbol{\nu}]$. The signs of the check-to-bit node messages on the edge $(i,j) \in \mathcal{E}$ are $\operatorname{sgn}(\mu_{i\leftarrow j}) = \prod_{m\in N(j)\setminus i} \operatorname{sgn}(\mu_{m\to j})$ and $\operatorname{sgn}(\nu_{i\leftarrow j}) = \prod_{m\in N(j)\setminus i} \operatorname{sgn}(\nu_{m\to j})$. Using this and the fact that $\operatorname{sgn}(\boldsymbol{\mu}) = \operatorname{sgn}(\boldsymbol{\nu})$, it follows that

$$\operatorname{sgn}(\mu_{i\leftarrow j}) = \operatorname{sgn}(\nu_{i\leftarrow j}) \tag{A.19}$$

for all $(i,j) \in \mathcal{E}$.

Since $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ are both WMS-consistent, we know that $\operatorname{sgn}(\mu_{i\leftarrow j}) = \operatorname{sgn}(\mu_{i\leftarrow j'})$ and $\operatorname{sgn}(\nu_{i\leftarrow j}) = \operatorname{sgn}(\nu_{i\leftarrow j'})$ for all j, $j' \in N(i)$. Thus, for each $(i,\, j) \in \mathcal{E}$, $\mu'_{i\to j}$ and $\nu'_{i\to j}$ can be expressed as

$$\mu'_{i\to j} = \operatorname{sgn}(\mu_{i\leftarrow j}) \left(\operatorname{sgn}(\mu_{i\leftarrow j})\,\gamma_i + \frac{\delta}{d_{\mathrm{v}} - 1} \sum_{k\in N(i)\setminus j} \min_{m'\in N(k)\setminus i} |\mu_{m'\to k}|\right) \tag{A.20}$$

and

$$\nu'_{i\to j} = \operatorname{sgn}\left(\nu_{i\leftarrow j}\right)\left(\operatorname{sgn}\left(\nu_{i\leftarrow j}\right)\gamma_i + \frac{\delta}{d_{\mathrm{v}}-1}\sum_{k\in N(i)\backslash j}\min_{m'\in N(k)\backslash i}\left|\nu_{m'\to k}\right|\right). \tag{A.21}$$

Since $|\boldsymbol{\mu}| \succeq |\boldsymbol{\nu}| \succ \frac{\|\boldsymbol{\gamma}\|_\infty}{\delta}\mathbf{1}$, we have

$$\begin{aligned}\operatorname{sgn}\left(\mu_{i\leftarrow j}\right)\gamma_i + \frac{\delta}{d_{\mathrm{v}}-1}\sum_{k\in N(i)\backslash j}\min_{m'\in N(k)\backslash i}\left|\mu_{m'\to k}\right| \\ > \operatorname{sgn}\left(\mu_{i\leftarrow j}\right)\gamma_i + \|\boldsymbol{\gamma}\|_\infty \geq 0\end{aligned} \tag{A.22}$$

and

$$\begin{aligned}\operatorname{sgn}\left(\nu_{i\leftarrow j}\right)\gamma_i + \frac{\delta}{d_{\mathrm{v}}-1}\sum_{k\in N(i)\backslash j}\min_{m'\in N(k)\backslash i}\left|\nu_{m'\to k}\right| \\ > \operatorname{sgn}\left(\nu_{i\leftarrow j}\right)\gamma_i + \|\boldsymbol{\gamma}\|_\infty \geq 0.\end{aligned} \tag{A.23}$$

Hence, one can have

$$\left|\mu'_{i\to j}\right| = \operatorname{sgn}\left(\mu_{i\leftarrow j}\right)\gamma_i + \frac{\delta}{d_{\mathrm{v}}-1}\sum_{k\in N(i)\backslash j}\min_{m'\in N(k)\backslash i}\left|\mu_{m'\to k}\right|$$

and

$$\left|\nu'_{i\to j}\right| = \operatorname{sgn}\left(\nu_{i\leftarrow j}\right)\gamma_i + \frac{\delta}{d_{\mathrm{v}}-1}\sum_{k\in N(i)\backslash j}\min_{m'\in N(k)\backslash i}\left|\nu_{m'\to k}\right|.$$

By $|\boldsymbol{\mu}| \succeq |\boldsymbol{\nu}|$ and (A.19), we have $|\boldsymbol{\mu}'| \succeq |\boldsymbol{\nu}'|$.

Moreover, from (A.20)–(A.23), the signs of $\mu'_{i\to j}$ and $\nu'_{i\to j}$ satisfy $\operatorname{sgn}(\nu'_{i\to j}) = \operatorname{sgn}(\nu_{i\leftarrow j})$ and $\operatorname{sgn}(\mu'_{i\to j}) = \operatorname{sgn}(\mu_{i\leftarrow j})$. Since $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are WMS-consistent, it follows

that $\operatorname{sgn}(\nu'_{i\to j}) = \operatorname{sgn}(\nu_{i\leftarrow j}) = \operatorname{sgn}(\nu_{i\to j})$ and $\operatorname{sgn}(\mu'_{i\to j}) = \operatorname{sgn}(\mu_{i\leftarrow j}) = \operatorname{sgn}(\mu_{i\to j})$ for all $(i,j) \in \mathcal{E}$. This concludes the proof.

A.9 Proof of Lemma 63

Before going into the proof of Lemma 63, we first introduce the following auxiliary lemma.

Lemma 71. *Given a received LLR vector $\boldsymbol{\gamma}$ and a $\delta \in [0,1]$, let $\mu_{i\to j}^{(\ell)}$ be the WMS message with $\beta = \frac{\delta}{d_{\mathrm{v}}-1}$ on the edge (i,j) in the ℓ-th iteration. For all $(i,j) \in \mathcal{E}$, the absolute value of the WMS message is upper bounded by*

$$|\mu_{i\to j}^{(\ell-1)}| \leq \ell \|\boldsymbol{\gamma}\|_\infty. \tag{A.24}$$

Proof. Using the update equations (3.13) and (3.14), the absolute value of the message $\mu_{i\to j}^{(\ell)}$ for any $\ell > 0$ can be upper bounded by

$$\begin{aligned}
\left|\mu_{i\to j}^{(\ell)}\right| &= \left|\gamma_i + \beta \sum_{k\in N(i)\setminus j} \left(\prod_{m\in N(k)\setminus i} \operatorname{sgn}\left(\mu_{m\to k}^{(\ell-1)}\right) \right) \min_{m'\in N(k)\setminus i} \left|\mu_{m'\to k}^{(\ell-1)}\right| \right| \\
&\leq |\gamma_i| + \beta \left| \sum_{k\in N(i)\setminus j} \left(\prod_{m\in N(k)\setminus i} \operatorname{sgn}\left(\mu_{m\to k}^{(\ell-1)}\right) \right) \min_{m'\in N(k)\setminus i} \left|\mu_{m'\to k}^{(\ell-1)}\right| \right| \\
&\leq \|\boldsymbol{\gamma}\|_\infty + \beta \sum_{k\in N(i)\setminus j} \min_{m'\in N(k)\setminus i} \left|\mu_{m'\to k}^{(\ell-1)}\right| \\
&\leq \|\boldsymbol{\gamma}\|_\infty + \beta(d_{\mathrm{v}}-1) \left\|\boldsymbol{\mu}^{(\ell-1)}\right\|_\infty \\
&\leq \|\boldsymbol{\gamma}\|_\infty + \left\|\boldsymbol{\mu}^{(\ell-1)}\right\|_\infty .
\end{aligned}$$

Since the last upper bound is independent of (i,j), we have the following recursion

$$\left\|\boldsymbol{\mu}^{(\ell)}\right\|_\infty \leq \|\boldsymbol{\gamma}\|_\infty + \beta(d_{\mathrm{v}}-1) \left\|\boldsymbol{\mu}^{(\ell-1)}\right\|_\infty . \tag{A.25}$$

By applying (A.25) recursively and the fact that $\left\|\boldsymbol{\mu}^{(0)}\right\|_\infty = \|\boldsymbol{\gamma}\|_\infty$, the inequality (A.24) is obtained.

Having established Lemma 71, we can proceed and prove Lemma 63.

Proof. Let $\mu_{i\to j}^{(\ell)} = (\boldsymbol{\mu}_1^{(\ell)})_{i\to j}$ and $\mu_{\delta,i\to j}^{(\ell)} = (\boldsymbol{\mu}_\delta^{(\ell)})_{i\to j}$. From (3.13), the absolute value of the difference $\mu_{i\to j}^{(\ell)} - \mu_{\delta,i\to j}^{(\ell)}$ in the ℓ-th iteration can be written as

$$\left|\mu_{i\to j}^{(\ell)} - \mu_{\delta,i\to j}^{(\ell)}\right| = \left|\frac{1}{d_\mathrm{v}-1}\sum_{k\in N(i)\backslash j}\mu_{i\leftarrow k}^{(\ell-1)} - \frac{\delta}{d_\mathrm{v}-1}\sum_{k'\in N(i)\backslash j}\mu_{\delta,i\leftarrow k'}^{(\ell-1)}\right|. \tag{A.26}$$

By the triangle inequality, Equation (A.26) is upper bounded by

$$\frac{\delta}{d_\mathrm{v}-1}\sum_{k\in N(i)\backslash j}\left|\mu_{i\leftarrow k}^{(\ell-1)} - \mu_{\delta,i\leftarrow k}^{(\ell-1)}\right| + \frac{1-\delta}{d_\mathrm{v}-1}\sum_{k'\in N(i)\backslash j}\left|\mu_{i\leftarrow k'}^{(\ell-1)}\right|. \tag{A.27}$$

From (3.14) and Lemma 45, we know that $|\mu_{i\leftarrow k}^{(\ell-1)} - \mu_{\delta,i\leftarrow k}^{(\ell-1)}| \le \max_{m\in N(k)\backslash i}|\mu_{m\to k}^{(\ell-1)} - \mu_{\delta,m\to k}^{(\ell-1)}|$. Also, by the fact that $|\mu_{i\leftarrow k'}^{(\ell-1)}| = \min_{m\in N(k')\backslash i}|\mu_{m\to k'}^{(\ell-1)}|$, we can therefore upper bound (A.27) by

$$\begin{aligned}\frac{\delta}{d_\mathrm{v}-1}\sum_{k\in N(i)\backslash j}\max_{m\in N(k)\backslash i}\left|\mu_{m\to k}^{(\ell-1)} - \mu_{\delta,m\to k}^{(\ell-1)}\right| \\ + \frac{1-\delta}{d_\mathrm{v}-1}\sum_{k'\in N(i)\backslash j}\min_{m\in N(k')\backslash i}\left|\mu_{m\to k'}^{(\ell-1)}\right|.\end{aligned}$$

By Lemma 71 and the fact that $|\mu_{i\to j}^{(\ell-1)} - \mu_{\delta,i\to j}^{(\ell-1)}| \le \|\boldsymbol{\mu}^{(\ell-1)} - \boldsymbol{\mu}_\delta^{(\ell-1)}\|_\infty$ for all $(i,j)\in\mathcal{E}$, we have

$$\begin{aligned}\left|\mu_{i\to j}^{(\ell)} - \mu_{\delta,i\to j}^{(\ell)}\right| &\le \frac{\delta}{d_\mathrm{v}-1}\sum_{k\in N(i)\backslash j}\left\|\boldsymbol{\mu}^{(\ell-1)} - \boldsymbol{\mu}_\delta^{(\ell-1)}\right\|_\infty + \frac{1-\delta}{d_\mathrm{v}-1}\sum_{k'\in N(i)\backslash j}\ell\|\boldsymbol{\gamma}\|_\infty \\ &\le \left\|\boldsymbol{\mu}^{(\ell-1)} - \boldsymbol{\mu}_\delta^{(\ell-1)}\right\|_\infty + (1-\delta)\ell\|\boldsymbol{\gamma}\|_\infty. \end{aligned}\tag{A.28}$$

Since the RHS of (A.28) is a constant with respect to $(i,j) \in \mathcal{E}$, one gets the recursive upper bound

$$\left\| \boldsymbol{\mu}^{(\ell)} - \boldsymbol{\mu}_{\delta}^{(\ell)} \right\|_{\infty} \leq \left\| \boldsymbol{\mu}^{(\ell-1)} - \boldsymbol{\mu}_{\delta}^{(\ell-1)} \right\|_{\infty} + (1-\delta)\ell \|\boldsymbol{\gamma}\|_{\infty}. \tag{A.29}$$

Note that $|\mu_{i\to j}^{(0)} - \mu_{\delta,i\to j}^{(0)}| = 0$. For a given $\ell \leq L$, we can apply (A.29) recursively, and have

$$\begin{aligned} \left\| \boldsymbol{\mu}^{(\ell)} - \boldsymbol{\mu}_{\delta}^{(\ell)} \right\|_{\infty} &\leq (1-\delta)\frac{\ell\,(\ell+1)}{2}\|\boldsymbol{\gamma}\|_{\infty} \\ &\leq (1-\delta)\frac{L\,(L+1)}{2}\|\boldsymbol{\gamma}\|_{\infty}, \end{aligned}$$

for all $\ell \leq L$. Therefore, for any fixed $\epsilon > 0$, let

$$\delta_0(L) \triangleq 1 - \frac{2\epsilon}{L\,(L+1)\,\|\boldsymbol{\gamma}\|_{\infty}}. \tag{A.30}$$

For any $\delta \in [\delta_0(L), 1]$, we know that $|\mu_{i\to j}^{(\ell)} - \mu_{\delta,i\to j}^{(\ell)}| \leq \|\boldsymbol{\mu}^{(\ell)} - \boldsymbol{\mu}_{\delta}^{(\ell)}\|_{\infty} \leq \epsilon$ for all $\ell \leq L$

APPENDIX B

SUPPLEMENTAL MATERIALS

B.1 Extensions of the Work in [1]

In this appendix, we briefly recall the main idea and statement in our earlier work in [74], and provide detailed proof of lemmas which were omitted in [74]. We extend the lemmas and theorems in [1] to the case when the depth of the computation tree exceeds $\frac{1}{2}\text{girth}(\mathcal{G})$. With these extended results, another proof of the conclusion drawn in Section 3.3.2 is obtained.

Since a computation tree with depth greater than $\frac{1}{2}\text{girth}(\mathcal{G})$ is considered in this section, we generalize the definition in Section 3.2.2 as follows. Let $\mathcal{T}_{i_0}^{2T} = (\mathcal{I} \cup \mathcal{J}, \mathcal{E}')$ be a depth-$2T$ computation tree and rooted at $i_0 \in \mathcal{V}_\text{v}$, where $\mathcal{I}$ and $\mathcal{J}$ are the set of variable nodes and the set of check nodes in $\mathcal{T}_{i_0}^{2T}$, respectively, and $T \geq \frac{1}{4}\text{girth}(\mathcal{G})$. Let $i^{'}$ and $j^{'}$ denote a variable node and a check node in $\mathcal{T}_{i_0}^{2T}$, respectively. We say that i' is associated with the bit $i \in \mathcal{V}_\text{v}$ in $\mathcal{G}$ (denoted $i' \sim i$) if i' is a copy of $i \in \mathcal{V}_\text{v}$. Similarly, $j' \sim j$ denotes that $j' \in \mathcal{J}$ is a copy of $j \in \mathcal{V}_\text{c}$. Moreover, we define two projections $\eta : \mathcal{I} \to \mathcal{V}_\text{v}$ and $\theta : \mathcal{J} \to \mathcal{V}_\text{c}$ by $\eta(i^{'}) = \{i \in \mathcal{V}_\text{v} : i^{'} \sim i\}$ and $\theta(j^{'}) = \{j \in \mathcal{V}_\text{c} : j^{'} \sim j\}$. Note that the result of these maps, η and θ, are a subsets of $\mathcal{V}_\text{v}$. However, according to the specification, they are all singleton sets.

First, we generalize the definitions from [55] and [1, Definition 1] as follows.

Definition 72. Consider a computation tree $\mathcal{T}_{i_0}^{2T} = (\mathcal{I} \cup \mathcal{J}, \mathcal{E}')$ of depth $2T \geq \frac{1}{2}\text{girth}(\mathcal{G})$ rooted at i_0. A bit assignment $\boldsymbol{u} \in \{0,1\}^{|\mathcal{I}|}$ on $\mathcal{T}_{i_0}^{2T}$ is a *generalized valid deviation* of depth T at $i_0 \in \mathcal{V}_\text{v}$ or, in short, a *generalized T-local deviation* at i_0, if $u_{i_0} = 1$ and $\boldsymbol{u}$ satisfies all parity checks in $\mathcal{T}_{i_0}^{2T}$. Moreover, $\boldsymbol{u}$ is a *generalized minimal*

T-local deviation if, for every check node $j \in \mathcal{T}_{i_0}^{2T}$, at most two neighbor bits are assigned the value 1. Note that a generalized minimal T-local deviation at i_0 can be seen as a subtree of $\mathcal{T}_{i_0}^{2T}$ of depth $2T$ rooted at i_0, where every variable node has full degree and every check node has degree 2. Such a tree is referred as a *skinny tree.* If $\boldsymbol{\varpi} = (\varpi_0, \ldots, \varpi_T) \in [0,1]^T$ is a weight vector and $\boldsymbol{u}$ is a generalized minimal T-local deviation at i_0, then $\boldsymbol{u}^{(\boldsymbol{\varpi})}$ denotes the $\boldsymbol{\varpi}$*-weighted deviation*

$$u_i^{(\boldsymbol{\varpi})} = \begin{cases} \varpi_t u_i & \text{if } i \in N(i_0, 2t) \text{ and } 0 \le t \le T, \\ 0 & \text{otherwise,} \end{cases}$$

where $N(i_0, 2t)$ is the set of vertices in the $2t$-th level of $\mathcal{T}_{i_0}^{2T}$. For any $\boldsymbol{\varpi}$-weighted deviation $\boldsymbol{u}^{(\boldsymbol{\varpi})}$ on $\mathcal{T}_{i_0}^{2T}$, let the projection of $\boldsymbol{u}^{(\boldsymbol{\varpi})}$ onto the code bit $i \in \mathcal{V}_{\mathrm{v}}$ be

$$\begin{aligned} \pi_i\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right) &\triangleq \sum_{m' \in \mathcal{I}: m' \sim i} u_{m'}^{(\boldsymbol{\varpi})} \\ &= \sum_{t=0}^{T} \varpi_t \sum_{m' \in N(i_0, 2t): m' \sim i} u_{m'}. \end{aligned}$$

Likewise, we let $\boldsymbol{\pi}\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right)$ represent the vector whose elements are $\pi_i\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right)$ for $i \in \mathcal{V}_{\mathrm{v}}$. In the following, the weights are chosen to be $\varpi_t = \beta^t$ for some $\beta \in [0,1]$.

To extend the results of [1] to the computation trees of depth independent of girth($\mathcal{G}$), we utilize the following fact that, given the LLR vector $\boldsymbol{\gamma}$ and for each $i_0 \in \mathcal{V}_{\mathrm{v}}$, the WMS algorithm computes the best assignment, $\tilde{x}_{i_0}^{(T)}$, for the root of $\mathcal{T}_{i_0}^{2T}$. Also, there is a corresponding best assignment $\tilde{\boldsymbol{x}}^{(T)}$ for the tree $\mathcal{T}_{i_0}^{2T}$ that maximizes (3.9). Recall that $\mathcal{C}_{\mathcal{T}_{i_0}^{2T}}(x)$ is the set of all valid assignment of the tree $\mathcal{T}_{i_0}^{2T}$ with root assignment x, and $\mathcal{C}_{\mathcal{T}_{i_0}^{2T}} = \mathcal{C}_{\mathcal{T}_{i_0}^{2T}}(0) \cup \mathcal{C}_{\mathcal{T}_{i_0}^{2T}}(1)$. For a valid assignment

$\boldsymbol{x} \in \mathcal{C}_{\mathcal{T}_{i_0}^{2T}}$ and a $T' \leq T$, we define the function $V_{i_0}^{T'}(\boldsymbol{x})$ by

$$V_{i_0}^{T'}(\boldsymbol{x}) \triangleq \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T'} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} x_{m'}. \tag{B.1}$$

Let $\tilde{x}_{i_0}^{(T)}$ be the best root assignment of $\mathcal{T}_{i_0}^{2T}$, and let $\tilde{\boldsymbol{x}}^{(T)} \in \mathcal{C}_{\mathcal{T}_{i_0}^{2T}}(\tilde{x}_{i_0}^{(T)})$ maximize (3.9). We know

$$\begin{aligned}
\mu_{i_0}\left(\tilde{x}_{i_0}^{(T)}\right) &= \max_{\boldsymbol{x} \in \mathcal{C}_{\mathcal{T}}(\tilde{x}_{i_0}^{(T)})} \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} (1 - x_{m'}) \\
&= \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} \left(1 - \tilde{x}_{m'}^{(T)}\right) \\
&= \left(\sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} 1\right) - \left(\sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} \tilde{x}_{m'}^{(T)}\right) \\
&= V_{i_0}^{T}(\mathbf{1}) - V_{i_0}^{T}\left(\tilde{\boldsymbol{x}}^{(T)}\right).
\end{aligned} \tag{B.2}$$

The T'-level weighted correlation, denoted by $U_{i_0}^{T'}(\boldsymbol{x}, \boldsymbol{u})$, between an assignment $\boldsymbol{x}$ and a generalized minimal T-local deviation $\boldsymbol{u}$ for the tree $\mathcal{T}_{i_0}^{2T}$ is defined as

$$U_{i_0}^{T'}(\boldsymbol{x}, \boldsymbol{u}) \triangleq \sum_{i=0}^{n} \gamma_i \sum_{t=0}^{T'} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} (-1)^{x_{m'}} u_{m'}. \tag{B.3}$$

Let $\boldsymbol{x} \oplus \boldsymbol{u}$ be the modulo-2 sum of $\boldsymbol{x}$ and $\boldsymbol{u}$. Then, one can show that

$$\begin{aligned}
V_{i_0}^{T}(\boldsymbol{x} \oplus \boldsymbol{u}) &= \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} (x_{m'} + (-1)^{x_{m'}} u_{m'}) \\
&= \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} x_{m'} + \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0,2t): m' \sim i} (-1)^{x_{m'}} u_{m'} \\
&= V_{i_0}^{T}(\boldsymbol{x}) + U_{i_0}^{T}(\boldsymbol{x}, \boldsymbol{u}).
\end{aligned} \tag{B.4}$$

By the fact that $\tilde{\boldsymbol{x}}^{(T)}$ is the best assignment for the tree $\mathcal{T}_{i_0}^{2T}$, the following lemma shows that the weighted correlation is positive when the number of iterations is large enough.

Lemma 73. *Given the LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$ and a weight $0 < \beta < \frac{1}{d_{\mathrm{v}}-1}$, suppose the WMS algorithm converges to a WMS-consistent fixed point $\boldsymbol{\mu}^*$. For a $T > 0$, let $\tilde{\boldsymbol{x}}^{(T)} \in \mathcal{C}_{\mathcal{T}_{i_0}^{(2T)}}$ be the assignment such that the equality (B.2) holds and $\mu_{i_0}(\tilde{x}_{i_0}^{(T)}) - \mu_{i_0}(\tilde{x}_{i_0}^{(T)} \oplus 1) = |\mu_{i_0}^{(T)}|$. There exists an $T^* > 0$ such that, for all $T \geq T^*$ and for any generalized minimal T-local deviation $\boldsymbol{u}^{(T)}$ of the tree $\mathcal{T}_{i_0}^{2T}$, the weighted correlation $U_{i_0}^{T^*}(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}) > 0$.*

Proof. Let $\mu_{i \leftarrow j}^*$ be the converged check to bit message on the edge $(i, j) \in \mathcal{E}$, and let $\mu_i^* \triangleq \gamma_i + \beta \sum_{j \in N(i)} \mu_{i \leftarrow j}^*$. Similarly, we define $\mu_i^{(T)} \triangleq \gamma_i + \beta \sum_{j \in N(i)} \mu_{i \leftarrow j}^{(T)}$ for any $T > 0$. Let $\epsilon \triangleq \frac{1}{4} \min_{i \in \mathcal{V}_{\mathrm{v}}} |\mu_i^*|$. Since the WMS algorithm converges to $\boldsymbol{\mu}^*$, there exists a $T_1(\epsilon) > 0$ such that, for all $T \geq T_1(\epsilon)$ and $i \in \mathcal{V}_{\mathrm{v}}$, $|\mu_i^* - \mu_i^{(T)}| \leq \epsilon$. By the triangle inequality, one can show that $|\mu_i^{(T)}| \geq |\mu_i^*| - \epsilon > \epsilon$ for all $i \in \mathcal{V}_{\mathrm{v}}$. From the assumption of the lemma, we know that

$$\begin{aligned} \epsilon &< \mu_{i_0}\left(\tilde{x}_{i_0}^{(T)}\right) - \mu_{i_0}\left(\tilde{x}_{i_0}^{(T)} \oplus 1\right) \\ &\overset{\text{(a)}}{\leq} V_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)} \oplus \boldsymbol{u}^{(T)}\right) - V_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)}\right) \\ &\overset{\text{(b)}}{=} U_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right), \end{aligned} \tag{B.5}$$

where (a) holds because $\mu_{i_0}(\tilde{x}_{i_0}^{(T)} \oplus 1) \geq V_{i_0}^T(\mathbf{1}) - V_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)} \oplus \boldsymbol{u}^{(T)}\right)$ for all $\boldsymbol{u}^{(T)}$, and (b) follows from (B.4). From Definition 43, we know $\epsilon > 0$. Thus, we have $U_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) > 0$ when $T \geq T_1(\epsilon)$.

We further show that there exists a $T^* > 0$ such that $U_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) > 0$ for all

$T > T^*$ and all generalized minimal T-local deviation $\boldsymbol{u}^{(T)}$ of the tree $\mathcal{T}_{i_0}^{2T}$. Define

$$T_2(\epsilon) \triangleq \left\lceil \frac{\log\left(\epsilon - \epsilon\left(d_{\mathrm{v}} - 1\right)\beta\right) - \log\left(\|\boldsymbol{\gamma}\|_\infty d_{\mathrm{v}}\beta\right)}{\log\left(\left(d_{\mathrm{v}} - 1\right)\beta\right)} \right\rceil, \tag{B.6}$$

and let $T^* \triangleq \max\{T_1(\epsilon), T_2(\epsilon)\}$. For any $T > T^*$, we rewrite $U_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right)$ as

$$U_{i_0}^T\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) = U_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) + R_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right), \tag{B.7}$$

where

$$R^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) \triangleq \sum_{i=1}^{n} \gamma_i \sum_{t=T^*+1}^{T} \beta^t \sum_{m' \in N(i_0, 2t): m' \sim i} (-1)^{\tilde{x}_{m'}^{(T)}} u_{m'}^{(T)}. \tag{B.8}$$

By the fact that $R^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) \leq \left|R^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right)\right|$, the LHS of (B.8) can be upper bounded by,

$$\begin{aligned}
R^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) &\leq \sum_{i=1}^{n} |\gamma_i| \sum_{t=T^*+1}^{T} \beta^t \sum_{m' \in N(i_0, 2t): m' \sim i} u_{m'}^{(T)} \\
&\overset{\text{(a)}}{\leq} \sum_{i=1}^{n} |\gamma_i| \sum_{t=T_2(\epsilon)+1}^{T} \beta^t \sum_{m' \in N(i_0, 2t): m' \sim i} u_{m'}^{(T)} \\
&\leq \|\boldsymbol{\gamma}\|_\infty \sum_{i=1}^{n} \sum_{t=T_2(\epsilon)+1}^{T} \beta^t \sum_{m' \in N(i_0, 2t): m' \sim i} u_{m'}^{(T)} \\
&= \|\boldsymbol{\gamma}\|_\infty \sum_{t=T_2(\epsilon)+1}^{T} d_{\mathrm{v}}\left(d_{\mathrm{v}} - 1\right)^{t-1} \beta^t \\
&< \|\boldsymbol{\gamma}\|_\infty \sum_{t=T_2(\epsilon)+1}^{\infty} d_{\mathrm{v}}\left(d_{\mathrm{v}} - 1\right)^{t-1} \beta^t \\
&= \|\boldsymbol{\gamma}\|_\infty \frac{d_{\mathrm{v}}\beta}{1 - \left(d_{\mathrm{v}} - 1\right)\beta} \left(\left(d_{\mathrm{v}} - 1\right)\beta\right)^{T_2(\epsilon)},
\end{aligned} \tag{B.9}$$

where the inequality (a) follows from the fact that $T_2(\epsilon) \leq T^*$. By substituting (B.6)

into (B.9), we have $R^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) < \epsilon$. Moreover, by using (B.5) and (B.7), one can show

$$U_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) = U_{i_0}^{T}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) - R_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) > 0.$$

This completes the proof of the lemma.

Remark 74. Let $\tilde{\boldsymbol{x}}^{(T)}$, $\boldsymbol{u}^{(T)}$, and T^* be as defined in Lemma 73 with $T > T^*$. Since $U_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) > 0$, it follows that $V_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)} \oplus \boldsymbol{u}^{(T)}\right) = V_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}\right) + U_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}, \boldsymbol{u}^{(T)}\right) > V_{i_0}^{T^*}\left(\tilde{\boldsymbol{x}}^{(T)}\right)$ for all $\tilde{\boldsymbol{u}}^{(T)}$. This observation implies that, when $\beta < \frac{1}{d_{\mathrm{v}}-1}$ and the number of iterations is large, the binary assignments for the leaf nodes of the tree $\mathcal{T}_{i_0}^{2T}$ are asymptotically irrelevant to the best assignment of the root node $\tilde{x}_{i_0}^{(T)}$.

The following extends the key result [1, Lemma 4] to our generalized minimal local deviations on the computation tree.

Lemma 75. *Let $\mathcal{P}$ be the fundamental polytope of an LDPC, and $\boldsymbol{z} \in \mathcal{P}$ be an LP solution of a bit-regular code. Consider the set of depth-T computation trees with T independent of* girth$(\mathcal{G})$ *rooted at all non-zero variable nodes. For these trees, there exists a distribution over generalized minimal local deviations such that the expected value, when projected onto the original Tanner graph, is proportional to the LP solution $\boldsymbol{z}$.*

Proof. This fact was first observed in [77, Remark 22].

The following theorem shows that if the WMS messages converge to a WMS-consistent fixed point, then the hard decisions of the WMS algorithm define a codeword that is both LP and ML optimal.

Theorem 76. *Given a LLR vector $\boldsymbol{\gamma} \in \mathbb{R}^n$ and a weight $0 \leq \beta < \frac{1}{d_{\mathrm{v}}-1}$, suppose the WMS algorithm converges to a WMS-consistent fixed point. Then, the hard decisions,*

$\hat{\boldsymbol{x}}$*, form a* T*-locally optimal codeword [1, Definition 2] for some* T *independent of* girth$(\mathcal{G})$*. Moreover,* $\hat{\boldsymbol{x}}$ *is the LP optimal and, hence, ML codeword.*

Proof. From Theorem 44, we know that $\hat{\boldsymbol{x}}$ is a codeword. To prove that $\hat{\boldsymbol{x}}$ is a T-locally optimal codeword, we have to show that for the projection $\pi(\boldsymbol{u}^{(\varpi)})$ of any generalized minimal T-local deviation $\boldsymbol{u}^{(\varpi)}$, the inequality

$$\left\langle \hat{\boldsymbol{x}} \oplus c\pi\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right), \boldsymbol{\gamma} \right\rangle > \langle \hat{\boldsymbol{x}}, \boldsymbol{\gamma} \rangle$$

holds, where $c > 0$ is a scaling factor such that $c\pi_i(\boldsymbol{u}^{(\boldsymbol{\varpi})}) \leq 1$ for all $i \in 1, 2, \ldots, n$, and $(\hat{\boldsymbol{x}} \oplus c\pi(\boldsymbol{u}^{(\varpi)}))_i = |\hat{x}_i - c\pi_i(\boldsymbol{u}^{(\varpi)})|$ is as defined in [1]. Without loss of generality, we assume that $\boldsymbol{u}^{(\varpi)}$ is rooted at i_0 and consider the correlation of $\hat{\boldsymbol{x}} \oplus \pi(\boldsymbol{u}^{(\boldsymbol{\varpi})})$ and $\boldsymbol{\gamma}$. This gives

$$\begin{aligned}
\left\langle \hat{\boldsymbol{x}} \oplus c\pi\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right), \boldsymbol{\gamma} \right\rangle &= \sum_{i=1}^{n} \left| \hat{x}_i - c\pi_i\left(\boldsymbol{u}^{(\boldsymbol{\varpi})}\right) \right| \gamma_i \\
&= \langle \hat{\boldsymbol{x}}, \boldsymbol{\gamma} \rangle + c \sum_{i=1}^{n} \gamma_i \sum_{t=0}^{T} \beta^t \sum_{m' \in N(i_0, 2t):\, m' \sim i} (-1)^{\tilde{x}_{m'}} u_{m'} \\
&= \langle \hat{\boldsymbol{x}}, \boldsymbol{\gamma} \rangle + c U_{i_0}^{T}\left(\tilde{\boldsymbol{x}}, \boldsymbol{u}\right),
\end{aligned}$$

where $\tilde{\boldsymbol{x}} \in \mathcal{C}_{\mathcal{T}_{i_0}^{2T}}$ and $\tilde{x}_{i'} = \hat{x}_i$ if $i' \sim i$, and $U_{i_0}^{T}(\tilde{\boldsymbol{x}}, \boldsymbol{u})$ is defined in (B.3).

To show that $U_{i_0}^{T}(\tilde{\boldsymbol{x}}, \boldsymbol{u}) > 0$ for some $T > 0$, consider a tree $\mathcal{T}_{i_0}^{2T'}$ of depth $2T' = 2(T^* + T_1(\epsilon))$ rooted at $i_0 \in \mathcal{V}_{\mathrm{v}}$. Note that the constants ϵ, T^*, and $T_1(\epsilon)$ are defined in the proof of Lemma 73. Let $\tilde{\boldsymbol{x}}^{(T')}$ be the best assignment for the tree $\mathcal{T}_{i_0}^{2T'}$. Since we know that $\operatorname{sgn}\left(\mu_i^{(\ell)}\right) = \operatorname{sgn}\left(\mu_i^*\right)$ for all $\ell \geq T_1(\epsilon)$, the best assignment for each $i' \in \mathcal{T}_{i_0}^{2(T'-T_1(\epsilon))}$ is $\tilde{x}_{i'}^{(T')} = \hat{x}_i$, where $i' \sim i$. Here, Lemma 73 is required because the leaf assignment may not match a codeword. Also, $\boldsymbol{u}$ can be obtained from the

generalized minimal T'-local deviation $\boldsymbol{u}^{(T')}$ on $\mathcal{T}_{i_0}^{2T'}$ by truncating

$$u_{m'} = \begin{cases} u_{m'}^{(T')} & \text{if } m' \in N(i_0, 2t) \text{ for some } 0 \leq t \leq T' - T_1(\epsilon), \\ 0 & \text{otherwise.} \end{cases}$$

Let $T = T^*$. By Lemma 73, one can see that $U_{i_0}^T(\tilde{\boldsymbol{x}}, \boldsymbol{u}) = U_{i_0}^T(\tilde{\boldsymbol{x}}^{(T')}, \boldsymbol{u}^{(T')}) > 0$. Therefore, $\hat{\boldsymbol{x}}$ is a T-locally optimal codeword.

According to [1, Theorem 4] or [67, Theorem 6], and by Lemma 75, the T-local optimality of $\hat{\boldsymbol{x}}$ implies that $\hat{\boldsymbol{x}}$ is the unique optimal LP solution given the LLR $\boldsymbol{\gamma}$. Since $\hat{\boldsymbol{x}} \in \{0, 1\}^n$ is a codeword, $\hat{\boldsymbol{x}}$ is also an ML codeword.