Design and Analysis of Capacity-Achieving Codes and Optimal Receivers with Low Complexity

by

Chun-Hao Hsu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2006

Doctoral Committee:
Associate Professor Achilleas Anastasopoulos, Chair
Associate Professor Hendrikus G. Derksen
Associate Professor Serap Savari
Assistant Professor Sandeep P. Sadanandarao

To My Parents and My Wife

ACKNOWLEDGEMENTS

It is immeasurable how I have benefited from my thesis advisor Professor Achilleas Anastasopoulos throughout my doctoral study. He is incessantly encouraging, supportive, and ready to provide practical help and advice. The freedom and genuine enthusiasm I deeply enjoyed when working with him have always been the greatest nutrition for the growth of this thesis work. I have also learnt a lot from his genial personality, crispy communication skills and wisdom, which I will certainly carry beyond my PhD. My best gratitude goes to my best advisor.

I would like to thank Professors Hendrikus G. Derksen, Serap Savari and S. Sandeep Pradhan for their kind interest and willingness to serve in my thesis committee. I would also like to thank Chi-Yao Yu, Kai-Hsun Li, Chung-Yen Chao, Fang-Yi Chiang, Wen-Chiao Lin, and all other friends for their generous help, heartwarming friendship and the happy times we had together.

Finally, I would like to express my eternal gratefulness to my parents and my wife Sing-Rong. The unconditional love and support from my parents have fueled me with all the confidence and courage to complete my PhD. My wife Sing-Rong is always there supporting me, caring for me, bringing joy to my life, and even being my project teammate to make my life easier. Truly, this thesis would not be possible without them.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

APPENDIX

CHAPTER 1

Introduction

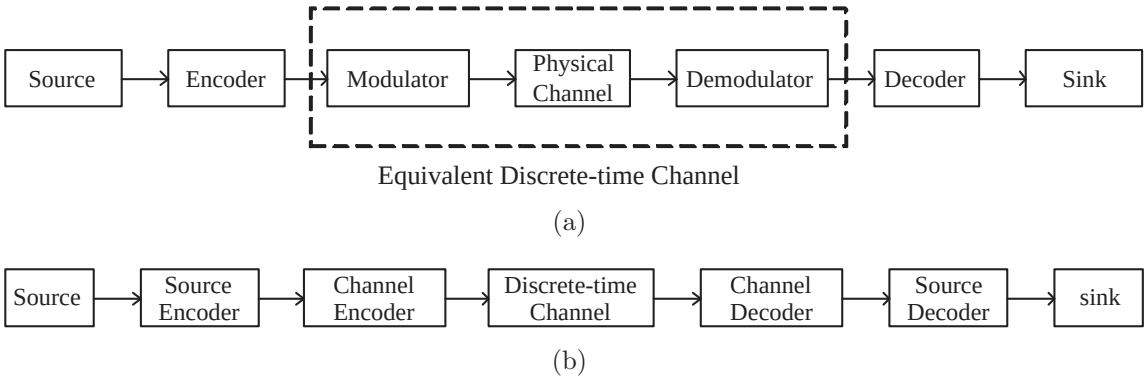


Figure 1.1: Digital communication system

Consider the point-to-point digital communication system as shown in Fig 1.1(a), where the information from a source is encoded, transmitted through a discrete-time noisy channel, and then decoded for an information sink. One of the most important tasks for communication engineers is to design the encoder and decoder for a highest possible transmission rate and a smallest possible distortion. In 1948, Shannon initiated the development and established in [1] the mathematical foundation of information theory, which shows that reliable communication with an arbitrarily small distortion is possible for a positive information transmission rate, highest of which is called the capacity of the channel. Moreover, in Shannon's source-channel separa-

tion theorem, it is shown that the optimal coding functionality can be accomplished separately by a source coder and a channel coder as shown in Fig. 1.1(b). The main objective of the source coder is to describe the source information by a sequence of symbols as efficiently as possible while preserving a high fidelity. On the other hand, the channel coder aims at guarding the symbols against the noise of the channel by properly and economically adding redundant symbols.

In this thesis, we focus on the channel coding problem, assuming as a common practice that every input symbol is statistically independent with each other and is chosen equiprobably among all possible symbols from the alphabet. This statistical assumption is practically well approximated when the source coder achieves a nearly optimal coding rate for any given tolerable distortion. Although Shannon's channel coding theorem (for memoryless channels) asserts that it is very easy to find capacity-achieving codes – just randomly construct it, then with asymptotically high probability as the codeword length N goes to infinity, we will find it – it does not take into account the encoding and decoding complexity of the code, which is indeed a limited resource in practical applications. Randomly constructed codes without structure can only employ a table-lookup algorithm for encoding and decoding, which soon becomes infeasible due to the exponential explosion of the cardinality of the codes as N increases. For this reason, we look at codes with structure. In particular, we restrict our attention to linear codes since linear codes admit a simple description by its generator matrix (or parity-check matrix) and can still achieve the channel capacity at least for the well-studied and commonly considered memoryless binary-input output-symmetric (MBIOS) channels[1].

The encoding complexity for linear codes is at most quadratic in N, whereas

[1]It is an easy exercise to show that the code ensemble specified by the parity-check matrix with entries being independently identically distributed (i.i.d.) Bernoulli random variables is capacity-achieving on any MBIOS channel using the Shulman-Feder bound given in [2].

the decoding complexity can vary depending on the decoding algorithm. To achieve capacity of a channel, conceivably a good decoding algorithm is required. Unfortunately, the optimal such algorithm, known as the maximum a-posteriori probability (MAP) decoding or equivalently the maximum-likelihood (ML) decoding under our statistical assumption on input symbols, turns out to be NP-complete[2] in its full generality, and NP-complete problems can only be solved with exponential complexity so far. This complexity demand is due to the memory imposed by the code structure and the channel, which prevents the problem of finding the most likely transmitted sequence from being solved by any efficient approaches but an exhaustive search in the whole codebook. Even in the case where the channel is memoryless, the general ML decoding problem for linear codes is still NP-complete [3]. Alternatively, when we consider uncoded transmission over channels with memory (e.g., fading channels with unknown channel state information at the transmitter and receiver), the problem of ML sequence detection (which in the above mentioned example is equivalent to an integer least squares problem) in general is NP-complete [4, 5] as well.

Despite the aforementioned negative facts for the channel coding problem, there is still hope of finding good coding schemes with a tractable complexity. For one, Shannon's channel coding theorem does not require an optimal decoder for a capacity-achieving code. A sub-optimal decoding algorithm might serve as well for a code to be capacity-achieving. Moreover, it is possible that for some specific channels with memory and code structures, the task of ML decoding is not as hard as in the general case. In the remaining of the thesis, we will concentrate on these two research directions under the following two scenarios.

[2]NP-complete problems are the computational problems which can be solved by a non-deterministic polynomial time algorithm. A non-deterministic algorithm is one which, when confronted with a choice between two alternatives, can create two copies of itself and simultaneously verify the correctness of these two copies. This repeated splitting may lead to an exponential growth of the number of copies.

1. When the channel is MBIOS, we design and analyze capacity-achieving codes defined on graphs, which admit a simple but powerful linear-complexity iterative message passing decoding.

2. When the channel has memory, we show that the ML decoding can be accomplished with polynomial complexity for some channels, both for the uncoded and simple trellis-coded transmissions.

1.1 Capacity-Achieving Codes Defined on Graphs

1.1.1 Background

In 1993, the remarkable discovery of turbo codes [6] with their associated iterative decoding phenomenally brought the best performance of known codes so close to the Shannon limit that probably no one could have expected. Since then, iterative decoding algorithms have attracted a large amount of attention, and been well understood as message passing algorithms defined on factor graphs [7]. In a factor graph, the transmitted bits of a code are represented as the visible variable nodes (or code nodes), the non-transmitted bits as hidden variable nodes (or state nodes), and the parity-check equations as check nodes. A code or state node is connected to a check node if the corresponding variable is involved in the corresponding parity check equation. An iterative message passing algorithm for a graph is one in which each node passes reliability messages to another node based on local observations, local code structure, and incoming messages in each iteration. If the graphical representation of a code is tree-like (or loop-free), then the sum-product algorithm [8], or equivalently the belief propagation (BP) algorithm, is guaranteed to find the optimal MAP decoding solution for each variable node. However, practical codes are generally loopy, in which case it is hard to analyze the BP decoding performance of

a code and to prove whether a code is capacity-achieving under BP decoding.

In [9], the irregular low-density parity-check (LDPC) codes are considered. Each irregular LDPC code ensemble is represented by a bipartite graph with code nodes and check nodes, and associated with a degree distribution pair (λ, ρ), where

$$\lambda(x) \triangleq \sum_{i=1}^{\infty} \lambda_i x^{i-1} \tag{1.1a}$$

$$\rho(x) \triangleq \sum_{i=1}^{\infty} \rho_i x^{i-1} \tag{1.1b}$$

specify the code and check node degree distributions, respectively. More precisely, λ_i (ρ_i, respectively) denotes the fraction of edges that are connected to a code (check) node with i check (code) node neighbors, assuming random permutation of edges between code and check nodes. The authors then proved the following two results for the irregular LDPC code ensembles on the binary erasure channel (BEC), which laid the foundation of and initiated the asymptotic performance analysis of codes under BP decoding.

Fact 1.1 (Concentration Around Ensemble Average) *Let G denote a particular code in the LDPC ensemble $\mathcal{C}(n, \lambda, \rho)$ with block length n and degree distribution pair (λ, ρ), and $P_b^{IT}(G, p, l)$ be the associated bit erasure probability if G is used to transmit over a BEC with erasure probability p and decoded by a BP decoder after l iterations. Then for any given $\delta > 0$, there exists an $\alpha(\delta) > 0$ such that*

$$Pr\{|P_b^{IT}(G, p, l) - E_{\mathcal{C}(n,\lambda,\rho)}[P_b^{IT}(G, p, l)]| > \delta\} \leq e^{-\alpha(\delta)n} \tag{1.2}$$

Fact 1.2 (Convergence of Ensemble Average to Loop-Free Case) *Under*

the same definitions as above, there exists a constant β such that

$$\left|E_{\mathcal{C}(n,\lambda,\rho)}[P_b^{IT}(G,p,l)] - E_{\mathcal{C}(\infty,\lambda,\rho)}[P_b^{IT}(G,p,l)]\right| \leq \frac{\beta}{n} \tag{1.3}$$

The first fact enables us to look at the average ensemble performance rather than the perforamance of some specific code in the ensemble, which greatly simplifies the analysis due to the randomness inherent in the construction of the ensemble. In addition, the second fact says that we can as well consider the case where the codeword length is so close to infinity that we would not see any loops within any finite number of iterations. Armed with these two results, several irregular LDPC ensembles have been proved to be capacity-achieving on the BEC using BP decoding [10–13]. Note that, LDPC codes, originally introduced by Gallager in 1963 [14], are the first provable capacity-achieving codes with linear decoding complexity.

The aforementioned two elementary facts were further generalized to MBIOS channels in [15], where the influential density evolution (DE) method was also introduced. The DE method tracks the probability density function (pdf) of messages exchanged between nodes for each iteration in a DE equation, and characterizes the successful decoding event for a bit as the event where the reliability message about the considered bit is true with perfect certainty. Consequently, the DE equation can be evaluated numerically to give the decoding capability of any LDPC codes on MBIOS channels. However, since the functional space of all pdf's is an infinite-dimensional space, the evaluation of the DE equation is generally an infinite-dimensional problem except on the BEC, which renders the DE method hardly applicable to the performance analysis of codes on channels other than the BEC. Hence, even though Fact 1.1, Fact 1.2 and the DE method were extended in [16] for the multi-edge type LDPC codes, which include almost all known codes defined on graphs, codes with BP decoding still can only be proved to be capacity-achieving on the BEC.

Successful examples other than the LDPC codes include the systematic irregular repeat-accumulate (IRA) codes [17] and the nonsystematic IRA codes [18]. Note that, there do exist provable capacity-achieving codes using other linear complexity decoding algorithms than BP on the binary symmetric channel (BSC), known as the expander codes [19]. However, their decoding complexity increases exponentially in $\frac{1}{\epsilon}$, where ϵ is the multiplicative gap to capacity[3], while that of the LDPC and IRA codes with BP decoding is conjectured [20, 21] to increase only like $\frac{1}{\epsilon}\ln\frac{1}{\epsilon}$.

1.1.2 Contributions of this Thesis

In this thesis, we devote Chapter 2 and Chapter 3 to the design and analysis of capacity-achieving codes defined on graphs under ML decoding on MBIOS channels, and also under BP decoding on the BEC. The motivation for looking at ML decoding is twofold. First, achieving capacity with ML decoding provides a necessary condition for achieving capacity with suboptimal message-passing decoding algorithms without resorting to the DE method, which, as mentioned earlier, is the main difficulty in extending the results from the BEC to MBIOS channels. Furthermore, it was shown in [22–25] that there are efficient methods to approach the ML decoding performance by improved iterative decoding algorithms. Thus, it is reasonable to treat the ML performance as a good indication of the real performance of codes even if an iterative decoding algorithm is used. As an ambitious attack to the performance analysis problem for codes with iterative decoding on MBIOS channels, we give tight iterative min-sum (MS) [8] and BP decoding performance bounds on MBIOS channels in Chapter 4, which also reveal some links between the iterative decoding performance of codes on MBIOS channels and that on the BEC. In the following three sections, we give brief introductions to the topics discussed in these three chapters.

[3]For a code operating reliably over a channel with capacity C, and having rate $R = (1-\epsilon)C$, its multiplicative gap to capacity is ϵ.

Capacity-Achieving Punctured LDPC Codes

For communication on time-varying channels (e.g., wireless channels), it is desirable to have one code whose rate and performance can be adapted according to the channel condition. This goal can be achieved by using punctured codes defined on graphs, where the code rate is changed by puncturing a subset of code bits. An advantage of using punctured codes defined on graphs is that it permits the use of a single encoder and (iterative) decoder for all punctured codes, in which case, rate-adaptability comes with no additional cost on encoding and decoding complexity. It is thus important to know how the performance of a code is affected by puncturing.

Another motivation for considering punctured codes is design-oriented. Since the capacity-achieving nonsystematic IRA codes can be viewed as punctured repeat-accumulate (RA) codes introduced in [26], which are non-capacity-achieving, we would like to ask the following question: "**Can capacity-achieving codes be more easily constructed, if we start from a code with lower rate, and then puncture it to the desired rate close to capacity?**"

In Chapter 2, we give a positive answer to the above question by proving that punctured LDPC codes can achieve capacity on MBIOS channels using ML decoding if they are punctured from some simple Gallager's LDPC codes [14] with low enough rate. Moreover, we prove that the multiplicative gap to capacity of all punctured codes can be the same as that of the original codes with low enough rate. These results are obtained by deriving and analyzing the average weight distribution (AWD) and its asymptotic growth rate of the punctured LDPC codes. Conditions under which puncturing results in only a negligible cardinality reduction of the original codebook with asymptotically high probability are also given in the process.

Capacity-Achieving Codes with Bounded Complexity

The complexity of iterative decoding, though linear in the codeword length, depends considerably on the complexity of the graphical representation of the codes. In particular, the iterative decoding complexity per iteration is directly proportional to the number of edges in the graph. Hence, the following question arises: "**How simple can the graphs be as a function of their performance?**"

In [27], the authors proved from an information theoretical perspective that if there are no state nodes in the graph, then the graphical complexity, i.e., the number of edges per information bit in the graph, should grow indefinitely as the multiplicative gap to capacity of the code decreases to zero on any MBIOS channel even when ML decoding is used. This rather discouraging result in terms of the performance-complexity tradeoff applies to the capacity-achieving LDPC and systematic IRA codes. On the other hand, by allowing state nodes in the graph, the authors in [18] were able to give two nonsystematic IRA ensembles that achieve capacity on the BEC using BP decoding with **bounded graphical complexity**. These results were obtained by analyzing the code performance using the DE method. However, partially due to the limitation of the DE method, whether graphs with state nodes can achieve capacity with bounded graphical complexity on more general channels other than the BEC still remains unknown.

Motivated by the above results, we first investigate the ML performance of nonsystematic IRA codes on MBIOS channels via deriving their AWD and its associate asymptotic growth rate. Numerical evaluation of these two metrics together with the use of Divsaler's ML performance bound [28] reveals that nonsystematic IRA codes have very good performance on the binary input additive white Gaussian noise (BI-AWGN) channel with a moderate graphical complexity. However, we were unable to analytically prove whether these codes can actually achieve capacity with bounded

graphical complexity. Motivated by the difficulty of analyzing nonsystematic IRA codes, we propose a new family of codes, called low-density parity-check and generator matrix (LDPC-GM) codes, that can achieve capacity on any MBIOS channel using ML decoding and also achieve capacity on the BEC using BP decoding, both with bounded graphical complexity. Properties of the LDPC-GM codes, which are constructed as serially concatenated codes with an outer LDPC code and an inner low-density generate matrix (LDGM) code [29], are also studied. In particular, we show that the proposed codes have linearly increasing minimum distances achieving the Gilbert-Varshamov bound for all code rates. In the process, the LDGM codes, which appear as an essential construction component in both the IRA and LDPC-GM codes, are also studied in detail.

Iterative Decoding Performance Bounds for LDPC Codes

As mentioned earlier, although the DE method can be used to numerically evaluate the exact asymptotic performance of codes with iterative decoding on MBIOS channels, the fact that the evolved densities in general require an infinite-dimensional description makes it unsuitable for the derivation of analytical performance bounds. In Chapter 4, we embark on solving this problem for LDPC codes with MS and BP decoding via two different approaches.

Due to Fact 1.2, the asymptotic probability of bit error can be interpreted as the probability of the root bit being in error of some tree code. When MS decoding is considered, we are able to bound this probability of error for the root bit by the sequence error probability of a subcode of the tree code assuming the transmission of the all-zero codeword. Further invoking the union bound on the sequence error probability, we obtain a recursive upper bound on the asymptotic bit error probability of LDPC codes after each iteration. This derived upper bound also holds for BP

decoding since the BP algorithm is optimal in minimizing the bit error probability of tree codes.

When we turn our attention to the BP decoding, we project the evolved densities to the probability of ML decoded bit errors and Bhattacharyya parameters, and track their evolutions. As a result, we obtain a recursive lower bound on the bit error probability and a recursive upper bound on the Bhattacharyya parameter, which is also an upper bound on the probability of bit error, after each iteration. More significantly, both recursions recover the one-dimensional DE equation for LDPC codes on the BEC with all inequalities satisfied by exact equalities. This further implies that the performance of LDPC codes under BP decoding on the BEC is the worst among all MBIOS channels with the same Bhattacharryya parameter, and is the best among all MBIOS channels with the same uncoded bit error probability. This link between the asymptotic BP decoding performance of LDPC codes on BEC and that on MBIOS channels also holds for the more general multi-edge type LDPC codes, including the IRA and LDPC-GM codes, since all our results stem from Fact 1.1 and Fact 1.2. Note that the recursive upper bound on the evolved Bhattacharyya parameters is also found in parallel in [30].

1.2 Polynomial-Complexity Optimal Receivers

1.2.1 Background

A variety of realistic wireless and wired channels have memory, i.e., the distortion caused by the channel on the transmitted signal is generally correlated from time to time. In this case, the optimal detection rule should take advantage of the memory of the channel, and thus can not be performed in a symbol-by-symbol manner. More specifically, if the channel state information (CSI) is unknown to the receiver, then

the optimal detection should be carried out based on the observation of the whole sequence, and is traditionally believed to require a computational complexity growing exponentially in the sequence length. This is equivalent to an exhaustive search in the set of all possible sequences.

1.2.2 Contributions of this Thesis

In Chapter 5 and Chapter 6, we challenge this traditional belief by showing that optimal detection can be performed in polynomial time on some channels for uncoded and simple trellis-coded sequences. The basic idea behind these results is to think of decision regions on the parameter space, whose dimension remains fixed, rather than the observation space, whose dimension grows with the sequence length, and is primarily motivated by the work in [31]. As a compromise between analytical simplicity and modelling accuracy, we consider block-independent channel models (originally introduced in [32]), where the channel parameters are considered to be fixed, but random, over a block of N symbols, and are independent from block to block. This assumption simplifies analysis by capturing the channel dynamics with a single parameter N and yet remains accurate for frequency-hopping and time-division multiple-access schemes. In the following, we give brief introductions to these two chapters.

Low Complexity Algorithms for Joint Data Detection and Frequency/Phase Estimation

We consider communication through a channel where the transmitted signal is first rotated by an unknown linear phase process and further corrupted by additive white Gaussian noise (AWGN). This model is intended to characterize the deteriorating effect of an unknown phase shift and frequency jitter, which are present in

wireless communication systems, especially the ones with mobile transmitting and receiving ends. The detection is assumed to invoke the generalized-likelihood ratio test (GLRT) [33] criterion, which is normally used in situations where the statistical model of the channel parameters does not exist or is unknown to the receiver. Note that, the GLRT solutions can coincide with the optimal maximum-likelihood sequence detection (MLSD) solutions under some statistical conditions, and still appear to demand an exponential computational complexity in the sequence length.

In Chapter 5, we give an algorithm that finds the exact GLRT solution with $O(N^4)$ worst-case complexity regardless of the operating signal-to-noise ratio (SNR). The concepts used in the proof of the polynomial complexity result are also utilized to evaluate tight performance bounds on the exact and a family of approximate algorithms. These analytical results are then verified by various simulations.

Polynomial-Complexity Optimal Decoding of Trellis Codes Transmitted through Fading Channels

Although we show in Chapter 5 that the parameter-space-partitioning methodology introduced in [31] is also applicable to the frequency/phase-jitter channel for uncoded sequences, it is not clear whether this methodology is also feasible for coded sequences.

In Chapter 6, we consider the problem of optimal decoding for trellis-coded sequences transmitted over a frequency non-selective, time-selective fading channel. In particular, we propose an algorithm, which is proved to find the optimal maximum a posteriori probability sequence detection (MAPSqD) solution, at least for two-state trellis codes, with worst-case polynomial complexity in N for any SNR. Important applications of two-state trellis codes include the differentially encoded binary phase shift keying (BPSK) system and the IRA codes, which can be decomposed as a serial

concatenation of an outer LDGM code and an inner accumulator code as shown in Chapter 3.

1.3 Dissertation Outline

The rest of the thesis is organized as follows. In Chapter 2, properties of punctured LDPC codes are derived and it is shown that they are capacity-achieving on MBIOS channels. We design and analyze capacity-achieving codes with bounded complexity on MBIOS channels in Chapter 3. The asymptotic iterative decoding performance of LDPC codes on MBIOS channels is analyzed in Chapter 4, where lower and upper performance bounds for MS and BP decoding are derived. We turn our attention to channels with memory and optimal detection for uncoded sequences in Chapter 5. After that, optimal detection of coded sequences on channels with memory is studied in Chapter 6. Finally, we conclude this thesis and discuss future research directions in Chapter 7. Appendix A, Appendix B, Appendix C, Appendix D, and Appendix E consist of proofs for results in Chapter 2, Chapter 3, Chapter 5, Chapter 6, and Chapter 7, respectively. The contents of this thesis have appeared in part in [34–41].

CHAPTER 2

Capacity-Achieving Punctured LDPC Codes

2.1 Introduction

Low-density parity-check (LDPC) codes were shown in [11–13] to achieve the capacity of the binary erasure channel (BEC) using iterative decoding, and also in [27] to achieve the capacity of any memoryless binary-input output-symmetric (MBIOS) channel using maximum-likelihood (ML) decoding. A necessary condition for these LDPC codes to be capacity-achieving as proved in [27] is that their average check node degrees should go to infinity as capacity is approached, i.e., the graphical representation of these LDPC codes becomes very complicated when we are close to capacity. In this chapter, we would like to explore another strategy for designing capacity-achieving codes without large check node degrees by considering punctured codes. More precisely, a potentially successful strategy for designing capacity-achieving codes is the following: **given any desired rate close to capacity, first construct a code with lower rate, and then puncture it to the desired rate**.

The fact that punctured codes can be good candidates for achieving capacity has been recognized by several researchers. In [42], the authors studied the perfor-

mance of punctured LDPC codes when used over the additive white Gaussian noise (AWGN) channel and showed numerically (using the Gaussian approximation [43] to density evolution) that puncturing only results in a small loss of multiplicative gap to capacity. Some analytical results on punctured LDPC codes can be found in [44], where the authors showed that any arbitrarily small gap to capacity for the original code can be preserved after puncturing on the BEC. Both of these works rely on the density evolution method [15], which becomes an intractable infinite-dimensional problem on channels other than the BEC, and thus cannot be used to prove whether punctured codes are capacity-achieving over more general MBIOS channels. In this chapter, we prove the feasibility of the aforementioned methodology for constructing capacity-achieving codes by showing that punctured LDPC codes can achieve the capacity of MBIOS channels using ML decoding when they are punctured from simple Gallager's LDPC codes with low enough rate.

In addition to the design purposes, puncturing also serves as a useful method to construct rate-adaptable codes for time-varying channels (e.g., wireless channels), in which case, the rate of the punctured code is adjusted according to the channel conditions by puncturing a subset of bits in the original code. If the decoder for the original code is compatible with the one for the punctured code (e.g., the commonly used belief propagation decoder), then the rate adaptability comes with no additional cost of encoding/decoding complexity. It is thus important to know how the performance of the punctured code changes with respect to the original code. In this chapter, we prove that punctured LDPC codes can preserve any arbitrarily small gap to capacity of the original codes under ML decoding via deriving and analyzing an upper bound on their average weight distributions (AWDs). Moreover, we derive conditions under which puncturing results in no rate reduction (with respect to the original codeword length) with asymptotically high probability in the

punctured LDPC ensemble, and prove that any desired rate in $(0, 1)$ can be achieved via puncturing without rate reduction if the rate of the original code is sufficiently small.

The remaining of this chapter is organized as follows. In Section 2.2, we review and prove some fundamental properties of Gallager's LDPC ensembles. Then, we introduce the punctured (Gallager's) LDPC ensembles, derive an upper bound on their AWD and guaranteed rate in Section 2.3. Detailed analysis of the upper bound on the AWD and its asymptotic growth rate of the punctured LDPC ensembles, as well as the main results of this chapter are given in Section 2.4. Finally, we conclude this chapter in Section 2.5.

2.2 Gallager's LDPC Ensemble

Consider Gallager's (n, j, k) LDPC ensemble as introduced in [14] with guaranteed rate $R_o = 1 - j/k$. Let $\overline{N_o(l)}$ be the average number of codewords of weight l in a randomly drawn code from the ensemble. The asymptotic growth rate of $\overline{N_o(l)}$ is given in [45] (it appears as an upper bound in [14]) to be

$$w_o(a) \triangleq \lim_{n\to\infty} \frac{1}{n} \ln \overline{N_o(an)} = (1 - R_o) \inf_{x>0} \left\{ \ln \frac{(1+x)^k + (1-x)^k}{2x^{ak}} \right\} - (j-1)H(a), \tag{2.1}$$

where $H(a) \triangleq -a \ln a - (1-a)\ln(1-a)$ is the binary entropy function evaluated with natural logarithms. Some useful characterizations of $\overline{N_o(l)}$ and $w_o(a)$ are summarized below.

Fact 2.1 *There exists a $\delta_o \in (0, 1/2)$, such that*

1. $\sum_{l\in(0,n\delta_o)} \overline{N_o(l)} = O(n^{-j+2})$.

2. *$w_o(a) < 0$ and has exactly one local minimum, but no local maximum for all $a \in (0, \delta_o)$.*

3. *$w_o(a) > 0$ for all $a \in (\delta_0, 1/2]$, and $w_o(\delta_o) = 0$.*

4. *$w_o(a)$ has exactly one local maximum at $a = 1/2$, and $w_o(1/2) = R_o \ln 2$.*

5. *When k is even, $\overline{N_o(l)} = \overline{N_o(n-l)}$, for all $l \in \{0, 1, \ldots, n\}$.*

In Fact 2.1, item 1 to 4 are either proved explicitly in [14, Appendix A] or direct results from there, and item 5 follows from the linearity of the LDPC codes and the fact that the all-1 word is always a codeword when k is even. In order to use item 5, and for other mathematical convenience, we will assume throughout this chapter that k is even. An example of $w_o(a)$ for $j = 5$ and $k = 10$ is depicted in Fig. 2.1.

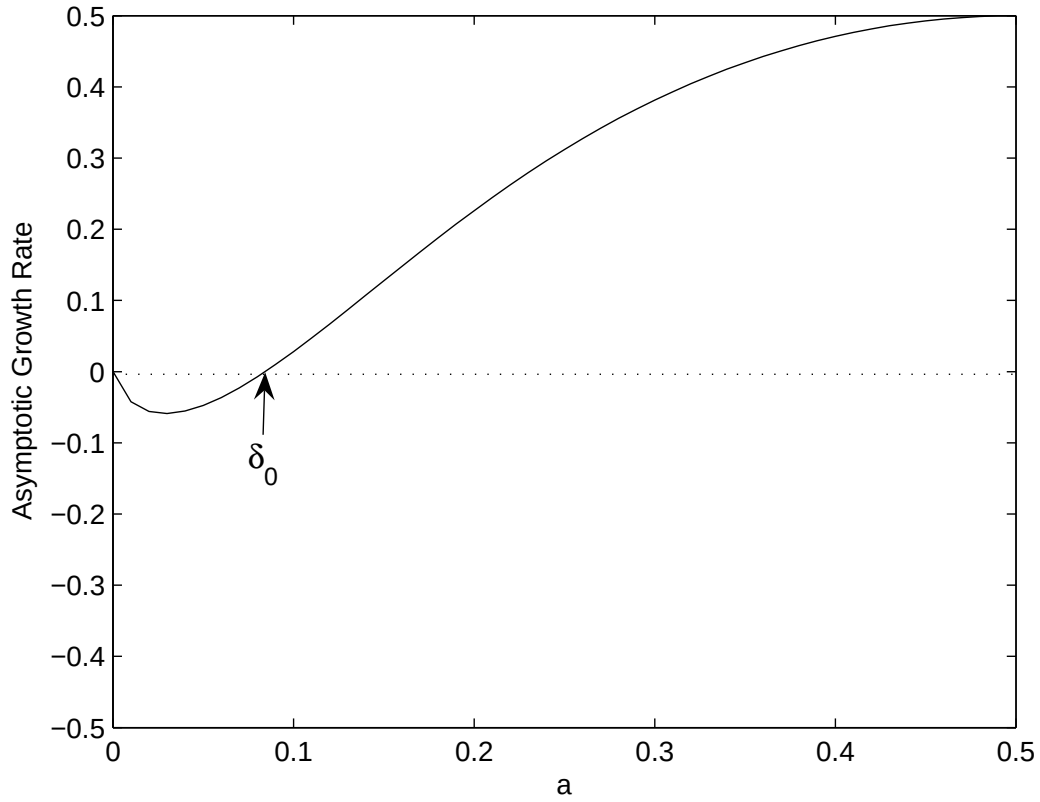


Figure 2.1: The asymptotic growth rate of the AWD of Gallager's (n, 5, 10) ensemble.

In the remaining of this section, we would like to prove several results, which will help our later analysis involving LDPC codes. First, we would like to give a close-form upper bound on $w_o(a)$, which is tight especially when a is around $1/2$.

Lemma 2.1 $w_o(a) \leq (1-R_o)\ln[1+(1-2a)^k] + [H(a) - (1-R_o)\ln 2] \triangleq w_o^{ub}(a)$.

Proof: Bounding the infimum term of (2.1) by substituting $x = \frac{a}{1-a}$, we have

$$\inf_{x>0}\left\{\ln\frac{(1+x)^k+(1-x)^k}{2x^{ak}}\right\} \leq \ln\frac{(1+x)^k+(1-x)^k}{2x^{ak}}\bigg|_{x=\frac{a}{1-a}} \tag{2.2a}$$

$$= \ln[1+(1-2a)^k] - \ln 2 + kH(a), \tag{2.2b}$$

from which the lemma follows straightforwardly. ■

Since the binary entropy function $H(x)$ appears in $w_o^{ub}(a)$ as shown in the previous lemma, it would facilitate our later analysis if we have a simple inequality on $H(x)$ as follows.

Lemma 2.2 *There exists a* $\delta_m \in [0, 1/2)$*, such that*

$$1-2a \leq \left(1 - \frac{H(a)}{\ln 2}\right)^{1/3}, \tag{2.3}$$

for all $a \in [\delta_m, 1/2]$.

Proof: Let $f(a) = [1-(1-2a)^3]\ln 2$. We have

$$\frac{d^2 f(a)}{da^2}\bigg|_{a=1/2} = 0 > -4 = \frac{d^2 H(a)}{da^2}\bigg|_{a=1/2}, \tag{2.4}$$

and

$$\frac{df(a)}{da}\bigg|_{a=1/2} = 0 = \frac{dH(a)}{da}\bigg|_{a=1/2}. \tag{2.5}$$

But, we also have

$$f(1/2) = \ln 2 = H(1/2). \tag{2.6}$$

Therefore, there must exist a $\delta_m \in [0, 1/2)$, such that

$$f(a) \geq H(a), \forall a \in [\delta_m, 1/2]. \tag{2.7}$$

Rearranging the above inequality proves the lemma. ■

In the following two lemmas, we give sufficient conditions for δ_o to be arbitrarily close to the normalized Gilbert-Varshamov distance, $H^{-1}((1 - R_o) \ln 2)$. The first way is to have a sufficiently large k as shown in the next lemma, where we denote by $H^{-1}(x)$ the unique $a \in [0, 1/2]$, such that $H(a) = x$.

Lemma 2.3 *Given any $\delta_l \in (0, H^{-1}((1 - R_o) \ln 2))$, if*

$$k > \frac{\ln\left[\left(1 - \frac{H(\delta_l)}{(1-R_o)\ln 2}\right)\ln 2\right]}{\ln(1 - 2\delta_l)}, \tag{2.8}$$

then $\delta_o > \delta_l$.

Proof: ¿From Lemma 2.1 we have

$$w_o^{ub}(\delta_l) < 0 \tag{2.9a}$$

$$\Leftrightarrow (1 - R_o) \ln[1 + (1 - 2\delta_l)^k] + [H(\delta_l) - (1 - R_o) \ln 2] < 0 \tag{2.9b}$$

$$\Leftrightarrow \ln[1 + (1 - 2\delta_l)^k] < \left(1 - \frac{H(\delta_l)}{(1 - R_o) \ln 2}\right) \ln 2 \tag{2.9c}$$

$$\Leftarrow (1 - 2\delta_l)^k < \left(1 - \frac{H(\delta_l)}{(1 - R_o) \ln 2}\right) \ln 2 \tag{2.9d}$$

$$\Leftrightarrow k > \frac{\ln\left[\left(1 - \frac{H(\delta_l)}{(1-R_o)\ln 2}\right)\ln 2\right]}{\ln(1 - 2\delta_l)}, \tag{2.9e}$$

where we have used the facts that $\ln(1+x) < x$, $\forall x \geq 0$ and $1 - 2\delta_l < 1$ in the last two steps. Now, the lemma follows from Lemma 2.1 and Fact 2.1. ■

Fig. 2.2 illustrates how the normalized Gilbert-Varshamov distance can be approached by increasing k for a fixed R_o.

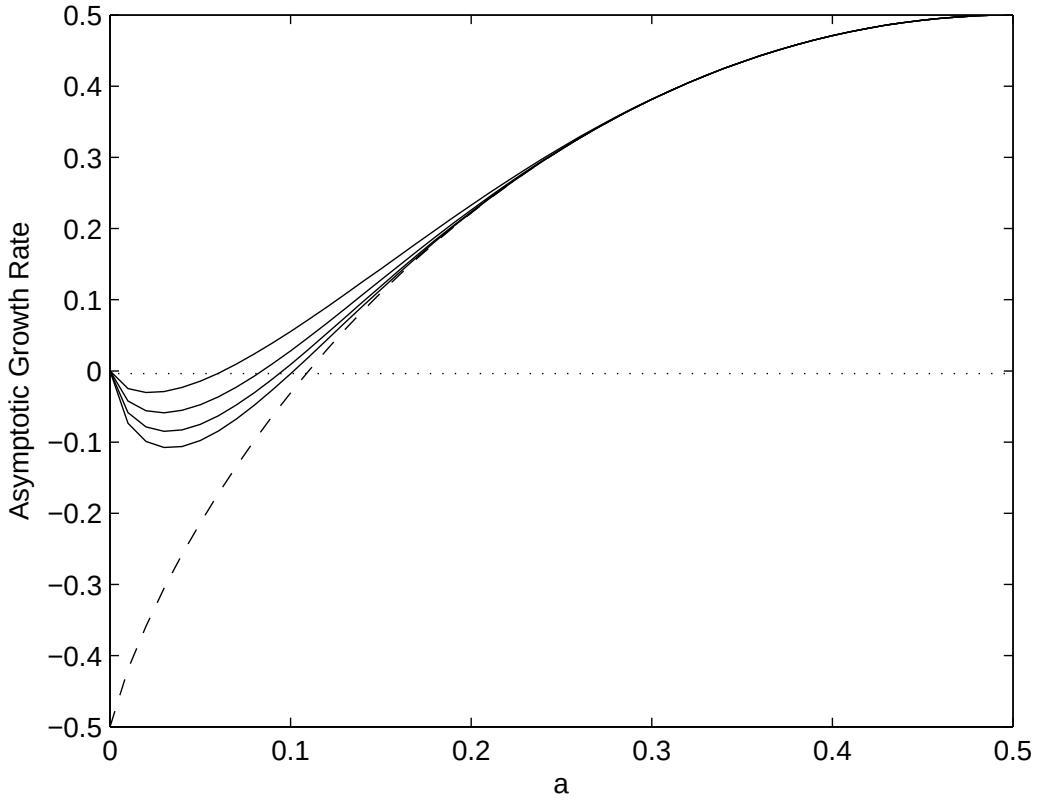


Figure 2.2: The dashed line depicts $H(a) - 0.5$, and solid lines depict $w_o(a)$ with $R_o = 0.5$, for $k = 8$, 10, 12 and 14 (from top to bottom), respectively. Note that the logarithms are to the base 2.

Alternatively, we can also keep a fixed $k \geq 4$, and make R_o sufficiently small for δ_o to be arbitrarily close to the normalized Gilbert-Varshamov distance. This is shown in the following lemma.

Lemma 2.4 *Given any $\eta > 1$ and $k \geq 4$, there exists an $S_1 > 0$, such that $\delta_o > H^{-1}((1 - \eta R_o)\ln 2)$ for all $R_o \in (0, S_1)$.*

Proof: Fix $\eta > 1$ and $k \geq 4$. Let $\delta_l = H^{-1}((1-\eta R_o)\ln 2)$ for all $R_o \in (0, 1/\eta)$. Then, there exists a $\xi \in (0, 1/\eta)$, such that $\delta_l \geq \delta_m$ for all $R_o \leq \xi$, where δ_m is as defined in Lemma 2.2. Therefore, for all $R_o \leq \xi$, we have from Lemma 2.2,

$$\frac{\ln\left[\left(1-\frac{H(\delta_l)}{(1-R_o)\ln 2}\right)\ln 2\right]}{\ln(1-2\delta_l)} \leq 3\frac{\ln\left[\left(1-\frac{H(\delta_l)}{(1-R_o)\ln 2}\right)\ln 2\right]}{\ln\left(1-\frac{H(\delta_l)}{\ln 2}\right)} \tag{2.10a}$$

$$= 3\frac{\ln\left[\frac{(\eta-1)R_o\ln 2}{1-R_o}\right]}{\ln(\eta R_o)} \tag{2.10b}$$

$$\leq 3\frac{\ln[(\eta-1)R_o\ln 2]}{\ln(\eta R_o)}, \tag{2.10c}$$

which approaches 3 monotonically as $R_o \to 0$. Note that the last inequality of (2.10) follows from the fact that $\ln(\eta R_o) < 0$. Hence, there exists an $S_1 \in (0, \xi]$, such that $3\frac{\ln[(\eta-1)R_o\ln 2]}{\ln(\eta R_o)} < 4$, for all $R_o < S_1$. This lemma then follows from Lemma 2.3. ■

Fig. 2.3 shows how the normalized Gilbert-Varshamov distance can be approached by decreasing R_o for a fixed k. We would like to point out that since j and k are integers, R_o in fact can not be arbitrarily small for any fixed $k \geq 4$. However, this restriction can be relaxed if we start from the following generalized Gallager's LDPC ensemble. We construct the parity-check matrix of an (n, j, k, γ) LDPC ensemble, where j and k are integers and $\gamma \in [0, 1)$ satisfies that k/γ is an integer, by appending a Gallager's $(n, 1, k/\gamma)$ parity-check matrix to a Gallager's (n, j, k) parity-check matrix (let the $(n, j, k, 0)$ ensemble be exactly the Gallager's (n, j, k) ensemble). Since there are totally $nj/k + n\gamma/k$ rows in an (n, j, k, γ) matrix, this ensemble has a guaranteed rate $R_o = 1-(j+\gamma)/k$, which can be made arbitrarily small by adjusting γ even for a fixed $k \geq 4$. Following similar steps as in [14], it can be verified that Fact 2.1 is still true for the generalized ensemble. Moreover, since $k/\gamma \geq k$ for all $\gamma \in [0, 1)$, it can be shown that Lemma 2.1 is also true for the generalized ensemble. Consequently, all the results derived in this chapter will still hold when the original ensemble is the generalized

LDPC ensemble, where we can safely treat k and R_o as two independent parameters. However, we chose to present all the results based on the original Gallager's ensemble mainly for the reason of not complicating the proofs with the additional details required for the case of the above mentioned generalized ensemble.

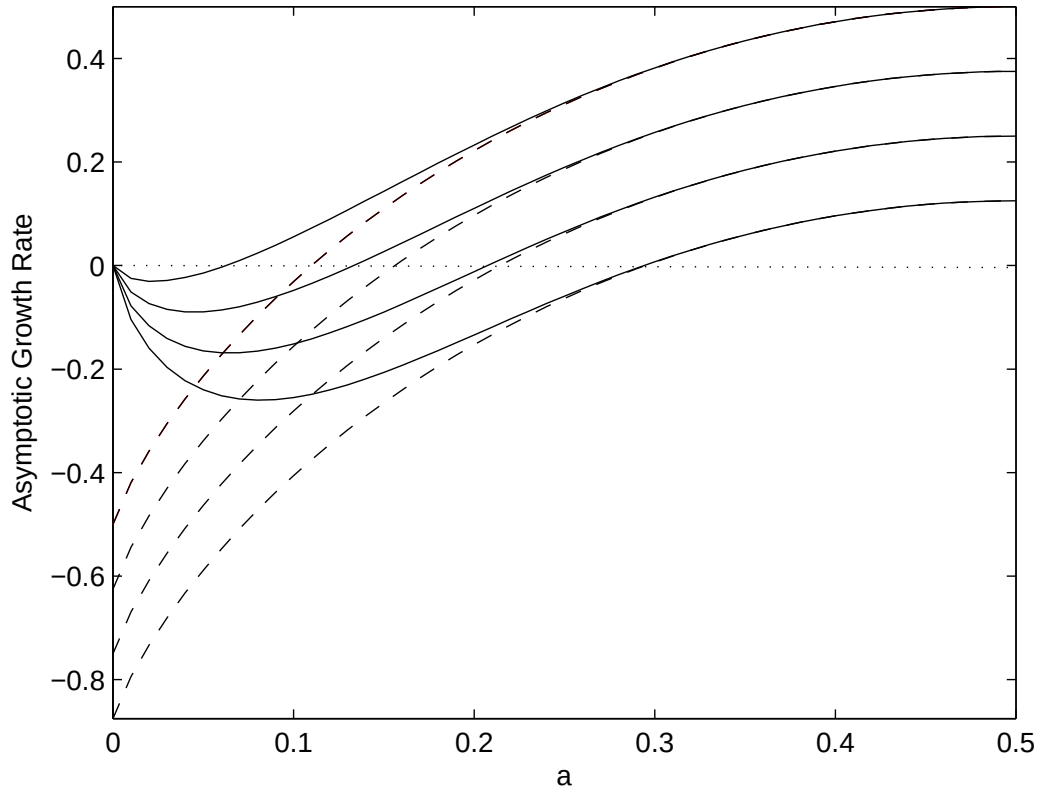


Figure 2.3: $H(a) - (1 - R_o)$ (dashed lines) and $w_o(a)$ (solid lines) with $k = 8$, for $R_o = 0.5$, 0.375, 0.25 and 0.125 (from top to bottom), respectively. Note that the logarithms are to the base 2.

2.3 Punctured LDPC Codes

Consider puncturing np bits of Gallager's (n, j, k) LDPC codes with rate R_o as described in the previous section. Note that, due to the randomness of the code construction of the LDPC codes, any properties of the punctured codes should not depend on the specific positions of the np punctured bits. We want to characterize

the AWD $\overline{N_p(l)}$, its asymptotic growth rate $w_p(a)$, and the resulting guaranteed rate R_p.

Let $\mathcal{C}_o$ be a randomly drawn code from the LDPC ensemble. If we ignore the possibility that different codewords in the original LDPC code can become the same codeword in the punctured code and overcount them, then we obtain the following upper bound on the AWD of the punctured ensemble

$$\begin{aligned}
\overline{N_p(l)} \leq & \overline{N_p^{ub}(l)} && \text{(2.11a)}\\
\triangleq & \sum_{i=0}^{pn} \left\{ \begin{array}{c} \text{average number of weight } i+l \text{ codewords} \\ \text{in the original ensemble whose } i \text{ 1's are in} \\ \text{the } np \text{ punctured positions and } l \text{ 1's are not} \end{array} \right\} && \text{(2.11b)}\\
= & \sum_{i=0}^{pn} \binom{pn}{i} \binom{(1-p)n}{l} P(\text{a weight } i+l \text{ word is in } \mathcal{C}_o) && \text{(2.11c)}\\
= & \sum_{i=0}^{pn} \frac{\binom{pn}{i}\binom{(1-p)n}{l}}{\binom{n}{i+l}} \overline{N_o(i+l)}. && \text{(2.11d)}
\end{aligned}$$

It then follows from (2.1) and the the well known property of binomial coefficients (see, e.g., [46, Lemma 18.9] for a proof)

$$\frac{1}{n} \ln \binom{n}{an} \to H(a) \text{ uniformly } \forall a \in [0,1] \text{ as n} \to \infty \tag{2.12}$$

that the asymptotic growth rate of this upper bound is given by

$$\begin{aligned}
w_p(a) \leq & w_p^{ub}(a) && \text{(2.13a)}\\
\triangleq & \lim_{n\to\infty} \frac{1}{(1-p)n} \ln \overline{N_p^{ub}(an)} && \text{(2.13b)}\\
= & H(a) + \frac{\max_{0\leq b\leq 1} T(a,b)}{1-p}, && \text{(2.13c)}
\end{aligned}$$

where

$$T(a,b) \triangleq pH(b) - H(pb + (1-p)a) + w_o(pb + (1-p)a) \tag{2.14}$$

To find the true rate R_p of the punctured codes, let $N_p^{ub}(0)$ be the random variable denoting the number of codewords in $\mathcal{C}_o$ which become the all-0 codeword in the punctured code $\mathcal{C}_p$. Then, due to the linearity of the LDPC codes, we have

$$\frac{|\mathcal{C}_o|}{N_p^{ub}(0)} = |\mathcal{C}_p|, \tag{2.15}$$

where $|\cdot|$ denotes the cardinality of some set. Hence, if we define

$$R' \triangleq \frac{1}{n} \log_2 \frac{|\mathcal{C}_o|}{|\mathcal{C}_p|} \tag{2.16}$$

to be the loss of rate with respect to the original codeword length n, then from Markov's inequality we have

$$P(R' \geq r) = P(N_p^{ub}(0) \geq 2^{nr}) \leq \frac{\overline{N_p^{ub}(0)}}{2^{nr}} \leq O\left(2^{n\left[\frac{(1-p)w_p^{ub}(0)}{\ln 2} - r\right]}\right), \tag{2.17}$$

which goes to 0 as n goes to infinity for all $r > \frac{(1-p)w_p^{ub}(0)}{\ln 2}$. Therefore, if we let

$$\Delta R \triangleq \max\left\{0, (1-p)w_p^{ub}(0)\right\} = \max\left\{0, \max_{0 \leq b \leq 1} T(0,b)\right\}, \tag{2.18}$$

then with asymptotically high probability, a guaranteed rate of the punctured codes can be given by

$$R_p = \frac{R_o - \Delta R / \ln 2}{1-p}. \tag{2.19}$$

Note that at this point, since $\overline{N_p^{ub}(l)}$ and $w_p^{ub}(a)$ are upper bounds of $\overline{N_p(l)}$ and

$w_p(a)$, respectively, they can be numerically evaluated and used to obtain various ML performance bounds as in [2, 28, 47] for the punctured ensemble. However, to obtain rigorously more analytical results and insight for the punctured LDPC codes, we would like to analyze punctured LDPC codes in more detail in the next section.

2.4 Analysis of the Punctured LDPC Codes

In this section, we first find conditions for $\Delta R = 0$, i.e., conditions for no rate reduction with asymptotically high probability, which implies $\overline{N_p^{ub}(l)}$ and $w_p^{ub}(a)$ are tight upper bounds. Then under these conditions, we further give upper bounds on $w_p^{ub}(a)$ and $\overline{N_p^{ub}(l)}$, and use them in conjunction with the ML decoding performance bound given in [48] to prove our main theorem.

Similar to what we showed in Lemma 2.3 and Lemma 2.4 that there are two ways for δ_o to be arbitrarily close to the normalized Gilbert-Varshamov distance, i.e., increasing k for a fixed R_o and decreasing R_o for a fixed $k \geq 4$, there are also two ways to achieve no rate loss, i.e., $\Delta R = 0$, after puncturing. The first way is to have a sufficiently large k as shown in the following lemma, and the second way is to have a sufficiently small R_o for a fixed $k \geq 4$ as shown in the consequent theorem.

Lemma 2.5 *If*

$$k > \frac{\ln[(1-\beta)\ln 2]}{\ln(1-2\delta_o)}, \tag{2.20}$$

where $\beta \triangleq \frac{p}{1-R_o}$, *then* $\Delta R = 0$.

Proof: We study the behavior of $T(0, b)$ in two cases. When $bp \in [0, \delta_o] \cup [1 -$

$\delta_o, 1]$, we have from Fact 2.1

$$\begin{aligned} T(0,b) =& pH(b) - H(pb) + w_0(pb) && \text{(2.21a)} \\ \leq& pH(b) - H(pb) && \text{(2.21b)} \\ \overset{(a)}{=}& -\left[pb \ln \frac{pb}{b} + (1-pb) \ln \frac{1-pb}{1-b}\right] + (1-p)\ln(1-b) && \text{(2.21c)} \\ \leq& (1-p)\ln(1-b) && \text{(2.21d)} \\ \leq& 0, && \text{(2.21e)} \end{aligned}$$

where the sum in the bracket following equality (a) is nonnegative since it is a relative entropy [49, Theorem 2.6.3]. On the other hand, when $bp \in (\delta_o, 1-\delta_o)$, we have from Lemma 2.1

$$\begin{aligned} T(0,b) \leq& pH(b) + (1-R_o)\ln[1 + (1-2bp)^k] - (1-R_o)\ln 2 && \text{(2.22a)} \\ \leq& p\ln 2 + (1-R_o)\ln[1 + (1-2bp)^k] - (1-R_o)\ln 2 && \text{(2.22b)} \\ =& (1-R_o)\{(\beta-1)\ln 2 + \ln[1 + (1-2bp)^k]\}. && \text{(2.22c)} \end{aligned}$$

Since $\ln[1 + (1-2x)^k]$ is a monotonically decreasing function in x, it attains its maximum at the left boundary, $x = \delta_o$, and we have

$$\max_{\delta_o \leq bp \leq 1-\delta_o} T(0,b) \leq (1-R_o)\{(\beta-1)\ln 2 + \ln[1 + (1-2\delta_o)^k]\}. \quad \text{(2.23)}$$

Therefore, from (2.21), (2.23) and (2.18),

$$\Delta R = 0 \Leftrightarrow \max_{0\le b\le 1} T(0,b) \le 0 \tag{2.24a}$$

$$\Leftarrow (1-R_o)\{(\beta-1)\ln 2 + \ln\left[1+(1-2\delta_o)^k\right]\} < 0 \tag{2.24b}$$

$$\Leftrightarrow \ln\left[1+(1-2\delta_o)^k\right] < (1-\beta)\ln 2 \tag{2.24c}$$

$$\Leftarrow (1-2\delta_o)^k < (1-\beta)\ln 2 \tag{2.24d}$$

$$\Leftrightarrow k > \frac{\ln[(1-\beta)\ln 2]}{\ln(1-2\delta_o)} \tag{2.24e}$$

where we have used the fact that $\ln(1+x) \le x$, $\forall x \ge 0$ in (2.24d), and the last step follows from the fact that $1-2\delta_o < 1$. ■

Lemma 2.5 implies that a sufficiently large k has to increase indefinitely to keep $\Delta R = 0$ as the puncturing probability p approaches $1 - R_o$. However, in the next theorem, we prove that even with a fixed $k \ge 4$, if the original ensemble has a small enough rate, then it can be punctured to any given set of rates with $\Delta R = 0$.

Theorem 2.1 *Given any $R_1 \in (0,1)$, and $k \ge 4$, there exists an $S_2 \in (0,1)$, such that if we have $R_o \in (0, S_2)$, then $\Delta R = 0$ for all $R_p \in [R_o, R_1]$.*

Proof: Fix $R_1 \in (0,1)$, and $k \ge 4$. From Lemma 2.4 and Lemma 2.2, we have for any $\eta > 1$, there exists an $S_1 > 0$, such that for all $R_o \in (0, S_1)$, $\delta_o > H^{-1}((1-\eta R_o)\ln 2) \ge \delta_m$. Thus, for any $R'_p \in [R_o, R_1]$, if we let $p = 1 - R_o/R'_p$, we

have

$$\frac{\ln[(1-\beta)\ln 2]}{\ln(1-2\delta_o)} \leq 3\frac{\ln\left[\left(1-\frac{1-R_o/R'_p}{1-R_o}\right)\ln 2\right]}{\ln\left(1-\frac{H(\delta_o)}{\ln 2}\right)} \tag{2.25a}$$

$$\leq 3\frac{\ln\left[\frac{(1/R'_p-1)R_o\ln 2}{1-R_o}\right]}{\ln(\eta R_o)} \tag{2.25b}$$

$$\leq 3\frac{\ln[(1/R_1-1)R_o\ln 2]}{\ln(\eta R_o)}, \tag{2.25c}$$

which approaches 3 monotonically as $R_o \to 0$. Therefore, there exists an $S_2 \in (0, S_1)$, such that $3\frac{\ln[(1/R_1-1)R_o\ln 2]}{\ln(\eta R_o)} < 4$, for all $R_o < S_2$. Hence, when $R_o \in (0, S_2)$, we have from Lemma 2.5 that $\Delta R = 0$ for all $k \geq 4$, and consequently

$$R_p = \frac{R_o - \Delta R/\ln 2}{1-p} = \frac{R_o}{R_o/R'_p} = R'_p, \tag{2.26}$$

which completes the proof of the theorem. ■

Knowing how to have $\Delta R = 0$, which implies tightness of the upper bounds $w_p^{ub}(a)$ and $\overline{N_p^{ub}(l)}$, we would like to proceed further to characterize $w_p^{ub}(a)$ and $\overline{N_p^{ub}(l)}$ in a way similar to Fact 2.1 as follows.

Theorem 2.2 *Given any $R_1 \in (0,1)$ and $k \geq 4$, choose an $R_o < S_2$, where S_2 is as given in Theorem 2.1 such that $\Delta R = 0$. Moreover, choose p so that $R_p \in [R_o, R_1]$. Let $n_p \triangleq (1-p)n$ be the codeword length of the punctured codes. Then there exists a $\delta_p \in (0, 1/2)$ such that $w_p^{ub}(a)$ and $\overline{N_p^{ub}(l)}$ satisfy the following properties:*

1. *$w_p^{ub}(0) \leq 0$.*
2. *$w_p^{ub}(a) < 0$ for all $a \in (0, \delta_p)$, and $w_p^{ub}(a) \leq \frac{(1-R_o)R_p}{R_o}\ln[1+(1-2\delta_o)^k] + [H(a) - (1-R_p)\ln 2]$ for all $a \in [\delta_p, 1/2]$.*
3. *$\overline{N_p^{ub}(l)} = O(n_p^{-j+3})$ for all $l \in (0, n_p\delta_p)$.*

4. $\overline{N_p^{ub}(l)} = \overline{N_p^{ub}(n_p - l)}$, *for all* $l \in [0, n_p]$.

Proof: See Appendix A.1. ■

Fig. 2.4 depicts $w_p^{ub}(a)$ for several punctured LDPC ensembles as examples. We would like to point out that item 3 of the above theorem in fact implies that with the choice of R_o as in the theorem and any $j \geq 5$, the punctured codes have a linearly increasing minimum distance with asymptotically high probability. Because, if we let d_{min} and $N_p(l)$ be the random variables denoting the minimum distance and number of codewords of weight l, respectively, of a randomly drawn code from the punctured ensemble, then from Markov's inequality, we have for all $\delta < \delta_p$,

$$\begin{aligned} P(d_{min} < \delta n_p) =& P\left(\sum_{l \in (0, \delta n_p)} N_p(l) \geq 1\right) && \text{(2.27a)} \\ \leq& \sum_{l \in (0, \delta n_p)} \overline{N_p(l)} && \text{(2.27b)} \\ \leq& n_p \max_{0 < l < \delta n_p} \overline{N_p^{ub}(l)} && \text{(2.27c)} \\ =& O\left(n_p^{-j+4}\right), && \text{(2.27d)} \end{aligned}$$

which goes to 0 asymptotically as n_p goes to infinity.

Armed with the upper bounds on $w_p^{ub}(a)$ and $\overline{N_p^{ub}(l)}$, we are now ready to state our main theorem, which shows that for any MBIOS channel, a capacity-achieving ensemble with any given rate can be constructed by puncturing any LDPC ensemble with $k > j \geq 5$ and a small enough rate. Moreover, it shows that the gap to capacity of all punctured LDPC codes with any rate $R_p \in [R_o, R_1]$, where R_o is small enough for no rate reduction for any target rate R_1 as implied in Theorem 2.1, is the same and decreases in $O(R_o^k)$, which depends only on the parameters of the original codes. In other words, under any MBIOS channel, there always exists a small enough R_o such that puncturing preserves the gap to capacity of the original codes for all punctured

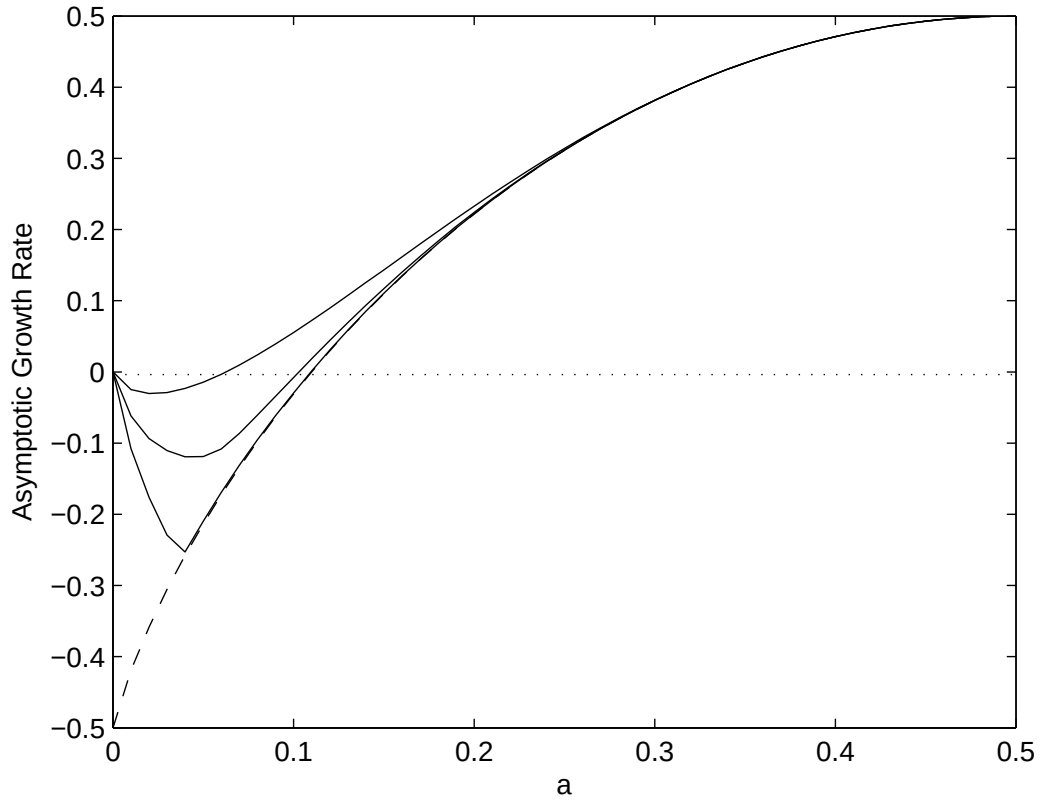


Figure 2.4: The dashed line depicts $H(a) - 0.5$, and solid lines depict $w_p^{ub}(a)$ with $k = 8$ and $\frac{R_o}{1-p} = 0.5$, for $R_o = 0.5$, 0.375, and 0.25 (from top to bottom), respectively. Note that the logarithms are to the base 2.

codes in any given rate range, which is consistent with the result in [44] for the BEC.

Theorem 2.3 *Given any $R_1 \in (0,1)$, and $k > j \geq 5$, let S_2 be as implied in Theorem 2.1. For any given $\eta > 1$, and $R_o \in (0, S_2)$, if $\epsilon > \eta^{k/3} R_o^{k/3-1}/\ln 2$, then the punctured ensemble with any rate $R_p \in [R_o, R_1]$ has a vanishing average block error probability under ML decoding on the MBIOS channel with capacity $C = \frac{R_p}{1-\epsilon}$.*

Proof: See Appendix A.2. ■

We conclude by making a point regarding the complexity of the punctured codes. A quantity of interest for codes defined on graphs, called the graphical complexity, is the number of edges per information bit in their graphical representation, which is directly proportional to their iterative decoding complexity per information bit per iteration. Although in the above theorem j and k can be kept fixed, it does not imply that the capacity of MBIOS channels can be achieved by punctured LDPC codes with a bounded graphical complexity. Indeed, the graphical complexity Δ of punctured LDPC codes can be calculated as follows.

$$\Delta = \frac{\text{\# of 1's in the p.c.m. of the original code}}{\text{number of information bits}} = \frac{nj}{n_p R_p} = \frac{nj}{n(1-p)R_o/(1-p)} = \frac{j}{R_o} \tag{2.28}$$

Therefore, as shown in Theorem 2.3, when the multiplicative gap to capacity ϵ approaches 0, we need R_o to approach 0 in order to achieve the channel capacity, which in turn implies that Δ approaches infinity. In particular, Δ should grow like $\epsilon^{-\frac{3}{k-3}}$ for a fixed k. This growth rate should be compared to the growth rate of $\ln \frac{1}{\epsilon}$ for capacity-achieving (unpunctured) LDPC codes reported in [27]. This comparison shows that although puncturing might be favorable for some values of ϵ away from 0, it is eventually not preferred for values of ϵ arbitrarily close to 0. The comparative advantage of the punctured codes on the other hand is their universality as demon-

strated in the last theorem. The inability of punctured codes to achieve capacity (even with ML decoding) with bounded complexity is the main motivation for the study reported in the following chapter.

2.5 Conclusion

Some fundamental properties of punctured LDPC codes are studied in this chapter. We prove that given any target rate R_1 in (0,1) and any $k \geq 4$, there always exists a small enough rate R_o of the original code, such that puncturing the original code to any rate less than R_1 is free of rate reduction with asymptotically high probability. In this case, the derived upper bounds on AWD and its asymptotic growth rate of the punctured LDPC codes become tight, and we analyze them in detail to prove three main results. First, punctured LDPC codes are "good", i.e., they have a linearly increasing minimum distance with asymptotically high probability. Second, punctured LDPC codes are capacity-achieving on any MBIOS channel under ML decoding if they are punctured from an original ensemble with a small enough rate but arbitrary $k > j \geq 5$. Third, for any given set of rates between 0 and 1, there always exists a small enough rate of the original ensemble, such that the gap to capacity of the original ensemble for any MBIOS channel is preserved for all punctured LDPC ensembles under ML decoding. These results show high potential for punctured LDPC codes to be used in rate compatible coding and design of capacity-achieving codes for general MBIOS channels.

CHAPTER 3

Capacity-Achieving Codes with Bounded Complexity

3.1 Introduction

In this chapter, we first investigate the maximum-likelihood (ML) decoding performance of irregular repeat-accumulate (IRA) codes on memoryless binary-input output-symmetric (MBIOS) channels. This code ensemble is chosen for the study of capacity-achieving codes on MBIOS channels with bounded graphical complexity, since the nonsystematic version of it is proved in [18] to have this property on the binary erasure channel (BEC). ML decoding performance analysis is performed via deriving the average weight distribution (AWD) of systematic and nonsystematic versions of the ensembles. The asymptotic growth rate of the AWD (in the following we refer to this quantity as the asymptotic average weight distribution (AAWD)) of IRA ensembles is also calculated, which can be used to obtain various ML decoding performance bounds as in [2, 28, 47]. In the process, the AAWD of low-density generator matrix (LDGM) ensembles [29] is also derived and used to prove that nonsystematic regular LDGM encoders, though are possible to, would not reduce

the information rate with asymptotically high probability. Furthermore, the role of the inner accumulator in spectral thinning is demonstrated. Our approach shows that simple nonsystematic IRA codes have a better guaranteed performance than systematic IRA and LDPC codes with the same graphical complexity, which is only 0.124 dB away from the Shannon limit when Divsalar's bound [28] is used on the binary-input additive white Gaussian noise (BIAWGN) channel. However, a conclusive answer as to whether these nonsystematic IRA ensembles achieve capacity was not reached. The reason lies in the fact that their AAWD cannot be proved to be strictly negative in the region of normalized weights close to zero. As a result, it cannot be guaranteed that the number of low weight codewords in these ensembles decreases exponentially fast. This further implies that their polynomial growth behavior has to be estimated; a seemingly more difficult task.

Motivated by the inconclusive result regarding the capacity-achieving property of IRA ensembles using ML decoding, we introduce a new family of codes, namely the low-density parity-check and generator matrix (LDPC-GM) codes, which are constructed by serially concatenating an outer LDPC code and an inner LDGM code. We prove that LDPC-GM codes can achieve capacity using ML decoding on any MBIOS channel with bounded graphical complexity. By deriving and analyzing the upper bounds on the AWD and AAWD of the LDPC-GM codes with a rate-1 LDGM inner code, we show that the inner rate-1 LDGM code helps eliminate high weight LDPC codewords while maintaining a vanishing small amount of low weight codewords. In addition to being capacity-achieving, it is also shown that these ensembles achieve the asymptotic Gilbert-Varshamov bound [50, 51]. As a supportive fact on the potential of these ensembles under iterative decoding, we also show that LDPC-GM ensembles can achieve capacity on the BEC with bounded decoding complexity (per information bit) for all erasure probabilities in $(0, 1)$. This

fact can also be viewed as an instance of symmetry observed in [52].

The remaining of this chapter is structured as follows. We derive and analyze the AWD and AAWD of IRA and LDGM codes in Section 3.2. The derived AAWD is then utilized to numerically evaluate the ML decoding performance of these codes on the BIAWGN channel. In Section 3.3, we introduce the LDPC-GM codes and prove that they are capacity-achieving with bounded graphical complexity on MBIOS channels. Allowing the outer LDPC code and inner LDGM code to be more generally irregular, we prove that the LDPC-GM codes can achieve capacity on any BEC with bounded decoding complexity in Section 3.4. Finally, we conclude this chapter in Section 3.5.

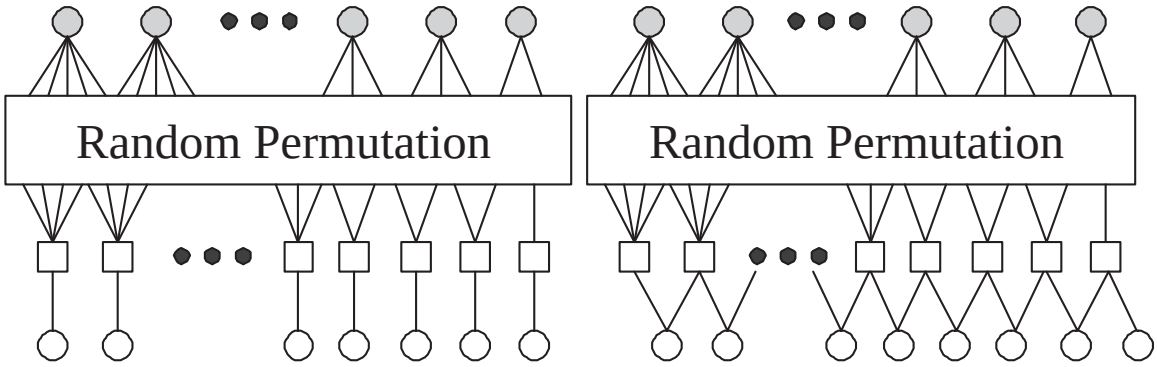
3.2 Average Weight Distribution of IRA Codes

One commonly used approach to analyze the ML performance of some code, is via deriving its input-output weight enumerator (IOWE) as in [26]. A general method for computing the IOWE of 1-input-t-output convolutional encoders is proposed in [53], which can then be used to analyze the ML performance of several concatenated code ensembles. However, if we view IRA codes as a serial concatenation of an outer repetition code and an inner convolutional code, then the inner convolutional encoder will have more than 1 input bits when the check node degree is greater than 3 as shown in Fig. 3.1(a). Therefore, the method in [53] unfortunately can not be directly applied to the general scenario of IRA ensembles with arbitrary check node degrees[1]. In this section, we solve this problem by viewing an IRA code as a serial concatenated code with an outer LDGM code and an inner accumulator code. Based on this decomposition, we derive the average input-parity weight enumerator (AIPWE) of

[1]However, when the check node degree is small, this problem can be circumvented by further decomposing the inner convolutional code into a regular check code and an accumulator code, and using the IOWE of check codes with small check degrees derived in [54].

IRA ensembles. Upper bounds on the AWD and AAWD of the systematic and nonsystematic versions of the IRA and LDGM ensembles are then obtained from their AIPWE, which can be used to obtain various ML performance bounds as in [2,28,47]. As an example, we use Divsalar's bound to compare the performances of systematic and nonsystematic IRA ensembles under ML decoding on the BIAWGN channel.

3.2.1 Background: LDGM and IRA Codes



(a) LDGM (b) IRA

Figure 3.1: Factor graph for LDGM and IRA codes. Information bits are denoted by filled gray circles, parity bits by open circles, and check nodes by squares.

Consider the LDGM and IRA codes as shown in Fig. 3.1. As can be seen in the figure, both of them have two different sets of variable nodes, i.e., the information nodes and the parity nodes. The systematic version of them uses all the variable nodes as its codeword, while the nonsystematic one uses only the parity nodes. Therefore, letting m denote the number of information bits and n denote the number of parity bits, the rate R of the systematic and nonsystematic codes is $m/(n+m)$ and m/n, respectively.

Let λ_i be the fraction of edges between the information and check nodes that are connected to an information node with i check node neighbors, and ρ_i be the fraction of the same edges that are connected to a check node with i information

node neighbors. Furthermore, define

$$\lambda(x) \triangleq \sum_{i=1}^{\infty} \lambda_i x^{i-1} \tag{3.1a}$$

$$\rho(x) \triangleq \sum_{i=1}^{\infty} \rho_i x^{i-1} \tag{3.1b}$$

to be the generating functions of λ_i's and ρ_i's. These two functions are used to specify the ensembles of LDGM and IRA codes assuming random permutation of edges between information and check nodes within each ensemble, and are known as the "degree distribution" pair. A special case is the "(c, d) regular" code ensemble defined by $\lambda(x) = x^{c-1}$ and $\rho(x) = x^{d-1}$.

The above degree distribution pair (λ, ρ) is from the edge perspective. It will facilitate our following analysis if we also have an equivalent description from the node perspective. Let $\tilde{\lambda}_i$ (respectively $\tilde{\rho}_i$) be the fraction of information (respectively check) nodes that are connected to i check (respectively information) nodes. Then we have

$$\tilde{\lambda}_i = \frac{\lambda_i / i}{\sum_{j=1}^{\infty} \lambda_j / j} \tag{3.2a}$$

$$\tilde{\rho}_i = \frac{\rho_i / i}{\sum_{j=1}^{\infty} \rho_j / j}. \tag{3.2b}$$

3.2.2 Average Input-Parity Weight Enumerator of LDGM and IRA Ensembles

The IOWE $A_{w,h}$ of a binary linear block code $\mathcal{C}$ is defined to be the number of codewords in $\mathcal{C}$ with input Hamming weight w and output Hamming weight h. Similarly, we can define the input-parity weight enumerator (IPWE) $Z_{w,h}$ for LDGM and IRA codes to denote the number of codewords with input weight w and parity

weight h. Note that the IPWE and IOWE are the same for nonsystematic LDGM and IRA codes, but different for systematic ones.

In this section, we calculate the AIPWE $\overline{Z_{w,h}}$ of LDGM and IRA ensembles, which is then used in the next section to obtain the AWD of systematic and nonsystematic versions of the respective ensembles.

AIPWE of LDGM Ensembles

Consider the (λ, ρ) LDGM ensemble. Let W and H be the random variables denoting the input and parity weight, respectively, of a randomly chosen codeword of a code drawn randomly from the ensemble. Furthermore, let E be the random variable denoting the total number of edges emanating from the information nodes that are equal to 1 in the aforementioned codeword. Moreover, define

$$t \triangleq m \sum_{i=1}^{\infty} i\tilde{\lambda}_i \tag{3.3}$$

to be the total number of edges between information and parity nodes. We have

$$\begin{aligned}
\overline{Z_{w,h}^{(LDGM)}} =& 2^k P(H = h, W = w) && \text{(3.4a)}\\
=& 2^k P(W = w) \sum_{e=0}^{t} P(H = h, E = e | W = w) && \text{(3.4b)}\\
=& \binom{k}{w} \sum_{e=0}^{t} P(H = h | E = e, W = w) P(E = e | W = w). && \text{(3.4c)}
\end{aligned}$$

The number of ways of having exactly e edges emanating from w information nodes, out of a total of $\binom{k}{w}$ possibilities, is equal to $\text{coef}(\prod_{u=1}^{\infty}(1 + x^u y)^{m\tilde{\lambda}_u}, x^e y^w)$, where $\text{coef}(f(x, y), x^a y^b)$ denotes the coefficient of $x^a y^b$ in the polynomial $f(x, y)$. Therefore,

we have

$$P(E=e|W=w)=\frac{\operatorname{coef}\left(\prod_{u=1}^{\infty}(1+x^u y)^{m\tilde{\lambda}_u}, x^e y^w\right)}{\binom{k}{w}}. \tag{3.5}$$

On the other hand, given that the number of edges from the information nodes equal to 1 is e, the output weight is h if and only if exactly h check nodes are connected to an odd number of such edges, and the remaining $n-h$ check nodes are connected to an even number of them. Counting the number of ways of connecting e edges to t check node sockets such that exactly h check nodes have an odd number of connections, we see that the value is equal to $\operatorname{coef}(\prod_{v=1}^{\infty}[f_-(x,v)y+f_+(x,v)]^{n\tilde{\rho}_v}, x^e y^h)$, where

$$f_-(x,v)\triangleq\frac{1}{2}[(1+x)^v-(1-x)^v] \tag{3.6a}$$

$$f_+(x,v)\triangleq\frac{1}{2}[(1+x)^v+(1-x)^v]. \tag{3.6b}$$

Since the total number of ways of connecting e edges to t sockets is equal to $\binom{t}{e}$, we have

$$P(H=h|E=e,W=w)=\frac{\operatorname{coef}\left(\prod_{v=1}^{\infty}[f_-(x,v)y+f_+(x,v)]^{n\tilde{\rho}_v}, x^e y^h\right)}{\binom{t}{e}}, \tag{3.7}$$

which is not related to the exact input weight w. Combining (3.4), (3.5) and (3.7), we obtain the AIPWE of the (λ,ρ) LDGM ensemble

$$\overline{Z_{w,h}^{(LDGM)}}= \sum_{e=0}^{t}\frac{1}{\binom{t}{e}}\operatorname{coef}\left(\prod_{u=1}^{\infty}(1+x^u y)^{m\tilde{\lambda}_u}, x^e y^w\right)\operatorname{coef}\left(\prod_{v=1}^{\infty}[f_-(x,v)y+f_+(x,v)]^{n\tilde{\rho}_v}, x^e y^h\right). \tag{3.8}$$

In particular, if $\lambda(x) = x^{c-1}$ and $\rho(x) = x^{d-1}$, we have $t = cm$, and

$$\operatorname{coef}\left((1+x^c y)^m, x^e y^w\right) = \begin{cases} \binom{m}{w} & \text{if } e = cw, \\ 0 & \text{else.} \end{cases} \tag{3.9}$$

$$\operatorname{coef}\left([f_-(x,d)y + f_+(x,d)]^n, x^e y^h\right) = \binom{n}{h} \operatorname{coef}\left(f_-(x,d)^h f_+(x,d)^{n-h}, x^e\right). \tag{3.10}$$

Hence (3.8) simplifies to the following AIPWE for the (c,d) regular LDGM ensemble

$$\overline{Z_{w,h}^{(LDGM)}} = \frac{\binom{m}{w}}{\binom{cm}{cw}} \binom{n}{h} \operatorname{coef}\left(f_-(x,d)^h f_+(x,d)^{n-h}, x^{cw}\right). \tag{3.11}$$

AIPWE of IRA Ensembles

A (λ, ρ) IRA code can be viewed as a serially concatenated code with an outer nonsystematic (λ, ρ) LDGM code and an inner accumulator code. In addition, the randomness of the LDGM ensemble construction is equivalent to having a uniform interleaver[2] [55] between the inner and outer codes. Based on the above, we have

$$\overline{Z_{w,h}^{(IRA)}} = \sum_{s=0}^{n} \frac{\overline{Z_{w,s}^{(LDGM)}} A_{s,h}^{(acc)}}{\binom{n}{s}}, \tag{3.12}$$

where $A_{w,h}^{(acc)}$ denotes the IOWE of the accumulator code, which is given in [26] to be

$$A_{w,h}^{(acc)} = \begin{cases} \binom{n-h}{\lfloor w/2 \rfloor} \binom{h-1}{\lceil w/2 \rceil - 1} & \text{if } \lfloor w/2 \rfloor \le n-h \text{ and } \lceil w/2 \rceil \le h, \\ 0 & \text{else.} \end{cases} \tag{3.13}$$

[2] A uniform interleaver of length n is a probabilistic device that maps any given input of weight w to all $\binom{n}{w}$ possible permutations of it with equal probability.

Hence, from (3.12), (3.13) and (3.8), we obtain the AIPWE of the (λ, ρ) IRA ensemble as

$$\overline{Z_{w,h}^{(IRA)}} = \sum_{\substack{s\geq 0, \lceil s/2\rceil \leq h \\ \lfloor s/2\rfloor \leq n-h}} \left\{ \frac{\binom{n-h}{\lfloor s/2\rfloor}\binom{h-1}{\lceil s/2\rceil -1}}{\binom{n}{s}} \right.$$
$$\left. \times \sum_{e=0}^{t} \frac{1}{\binom{t}{e}} \mathrm{coef}\left(\prod_{u=1}^{\infty}(1+x^u y)^{m\tilde{\lambda}_u}, x^e y^w\right) \mathrm{coef}\left(\prod_{v=1}^{\infty}[f_-(x,v)y + f_+(x,v)]^{n\tilde{\rho}_v}, x^e y^s\right)\right\}. \tag{3.14}$$

Similarly, from (3.12), (3.13) and (3.11), we obtain the AIPWE of the (c, d) regular IRA ensemble as

$$\overline{Z_{w,h}^{(IRA)}} = \frac{\binom{m}{w}}{\binom{cm}{cw}} \sum_{\substack{s\geq 0, \lceil s/2\rceil \leq h \\ \lfloor s/2\rfloor \leq n-h}} \binom{n-h}{\lfloor s/2\rfloor}\binom{h-1}{\lceil s/2\rceil -1} \mathrm{coef}\left(f_-(x,d)^s f_+(x,d)^{n-s}, x^{cw}\right). \tag{3.15}$$

3.2.3 Asymptotic Average Weight Distribution of LDGM and IRA Ensembles

Consider an LDGM or IRA ensemble with AIPWE $\overline{Z_{w,h}}$. Let $\overline{N(l)}$ be the average number of codewords of weight l in a randomly drawn code from the ensemble. Then for the systematic ensembles, their AWD is given by

$$\overline{N(l)} = \sum_{w=\max(0,l-n)}^{\min(m,l)} \overline{Z_{w,l-w}}. \tag{3.16}$$

However, for the nonsystematic ensembles, different input words can result in the same output codeword in a nonsystematic code. Such codewords should not be counted more than once in the weight distribution. As a result, the AWD of nonsystematic ensembles can not be obtained as directly as that for the systematic ones.

For example, for a nonsystematic regular LDGM code with even d, both the all-0 and the all-1 input word result in the all-0 output word. If we just overcount the repeated codewords in a code, then we obtain the following upper bound for the AWD of nonsystematic ensembles

$$\overline{N(l)} \leq \overline{N^{ub}(l)} \triangleq \sum_{w=0}^{m} \overline{Z_{w,l}}. \tag{3.17}$$

To analyze the asymptotic behavior of the AWD's, we use the following two equations proved in [56]

$$\lim_{\substack{n\to\infty \\ \mathrm{coef}(f(x),x^{an})\neq 0}} \frac{1}{n} \ln \mathrm{coef}(f(x)^n, x^{an}) = \inf_{x>0} \ln \frac{f(x)}{x^a} \tag{3.18a}$$

$$\lim_{\substack{n\to\infty \\ \mathrm{coef}(f(x,y),x^{an}y^{bn})\neq 0}} \frac{1}{n} \ln \mathrm{coef}(f(x,y)^n, x^{an}y^{bn}) = \inf_{x>0,y>0} \ln \frac{f(x,y)}{x^a y^b}, \tag{3.18b}$$

where $0 < a, b < 1$, and $f(x)$ and $f(x,y)$ are polynomials with nonnegative coefficients. Note that the convergence of (3.18a) ((3.18b), respectively) is uniform in a (and b) at the vicinity of any point such that $\inf_{x>0} \frac{f(x)}{x^a} \neq 0$ ($\inf_{x>0,y>0} \frac{f(x,y)}{x^a y^b} \neq 0$) as pointed out in [56]. The property of binomial coefficients as in (2.12) will also be used.

Define the AAWD of an ensemble with AWD $\overline{N(l)}$ to be

$$w(a) \triangleq \lim_{n\to\infty} \frac{1}{n} \ln \overline{N(an)}. \tag{3.19}$$

We have the following result.

Theorem 3.1 *The AAWD of the nonsystematic (c,d) regular LDGM, nonsystematic (c,d) regular IRA, systematic (c,d) regular LDGM, and systematic (c,d) regular*

IRA ensembles, respectively, satisfy

$$w(a) \leq H(a) + \max_{0\leq b\leq 1}\left\{R(1-c)H(b) + \inf_{x>0}\ln\frac{f_-(x,d)^a f_+(x,d)^{1-a}}{x^{bd}}\right\} \triangleq w^{ub}(a) \quad (3.20)$$

$$w(a) \leq \max_{0\leq r\leq \min(2(1-a),2a)}\left\{(1-a)H\left(\frac{r}{2(1-a)}\right) + aH\left(\frac{r}{2a}\right) + \max_{0\leq b\leq 1}\left[R(1-c)H(b) + \inf_{x>0}\log\frac{f_-(x,d)^r f_+(x,d)^{1-r}}{x^{bd}}\right]\right\} \triangleq w^{ub}(a) \quad (3.21)$$

$$w(a) = \max_{\max(0,\frac{a-1+R}{R})\leq b\leq \min(1,\frac{a}{R})}\left\{R(1-c)H(b) + (1-R)\left[H\left(\frac{a-bR}{1-R}\right) + \inf_{x>0}\log\frac{f_-(x,d)^{\frac{a-bR}{1-R}} f_+(x,d)^{1-\frac{a-bR}{1-R}}}{x^{bd}}\right]\right\} \quad (3.22)$$

$$w(a) = \max_{\max(0,\frac{a-1+R}{R})\leq b\leq \min(1,\frac{a}{R})} R(1-c)H(b) + \max_{0\leq r\leq \min(2(1-\frac{a-bR}{1-R}),2(\frac{a-bR}{1-R}))}\left\{(1-R)\left[\left(1-\frac{a-bR}{1-R}\right)H\left(\frac{r}{2\left(1-\frac{a-bR}{1-R}\right)}\right) + \frac{a-bR}{1-R}H\left(\frac{r}{2\left(\frac{a-bR}{1-R}\right)}\right) \inf_{x>0}\log\frac{f_-(x,d)^r f_+(x,d)^{1-r}}{x^{bd}}\right]\right\}, \quad (3.23)$$

where $H(a) \triangleq -a\ln a - (1-a)\ln(1-a)$ is the binary entropy function evaluated with natural logarithms.

Proof: From (3.17) and (3.11), the AAWD of the nonsystematic (c,d) regular

LDGM ensemble with $R \triangleq d/c$ and AWD $\overline{N(l)}$ can be calculated as follows[3]

$$w(a) = \lim_{n\to\infty} \frac{1}{n} \log \overline{N(an)} \tag{3.24a}$$

$$\leq \lim_{n\to\infty} \frac{1}{n} \log \overline{N^{ub}(an)} \tag{3.24b}$$

$$= \lim_{n\to\infty} \frac{1}{n} \ln \sum_{w=0}^{m} \overline{Z_{w,an}^{(LDGM)}} \tag{3.24c}$$

$$= \lim_{n\to\infty} \frac{1}{n} \ln \sum_{bm=0}^{m} \frac{\binom{m}{bm}\binom{n}{an}\text{coef}(f_-(x,d)^{an} f_+(x,d)^{(1-a)n}, x^{bcm})}{\binom{cm}{bcm}} \tag{3.24d}$$

$$= \lim_{n\to\infty} \frac{1}{n} \ln \left[\binom{n}{an} \max_{0\leq b\leq 1} \frac{\binom{Rn}{bRn}\text{coef}(f_-(x,d)^{an} f_+(x,d)^{(1-a)n}, x^{bdn})}{\binom{dn}{bdn}} \right] + o(n), \tag{3.24e}$$

where $o(n)$ is a function of n that converges to 0 as n approaches infinity. Now, (3.20) follows from (3.18a) and (2.12).

Similarly, (3.21), (3.22) and (3.23) can be derived from (3.11), (3.15), (3.16), (3.17), (3.18a) and (2.12). ■

Analogous results for the irregular LDGM and IRA ensembles can also be obtained in a straightforward manner from (3.8), (3.14), (3.16), (3.17), (3.18b) and (2.12), and are omitted here.

The derived upper bounds on the AAWD of nonsystematic ensembles are sufficient for obtaining various ML performance upper bounds. However, we still have to determine the guaranteed (design) rate of these codes due to the possible occurrence of rate reduction. In the following, we will show that regular nonsystematic LDGM and IRA codes indeed suffer no rate reduction with asymptotically high probability.

Let R_1 be the true rate of a randomly drawn code from the nonsystematic (c, d) regular LDGM ensemble. Let $N^{ub}(0)$ be the random variable denoting the number

[3] R may not be the true guaranteed rate of the nonsystematic (c, d) regular LDGM ensemble since nonsystematic codes with repeated codewords can undergo a rate reduction.

of input words that result in all-0 codeword in a randomly drawn code. Then, we have by the linearity of the LDGM codes and Markov's inequality that

$$\begin{aligned} P(R_1 < R - r) =& P(2^{nR}/2^{nR_1} > 2^{nr}) && \text{(3.25a)} \\ =& P(N^{ub}(0) > 2^{nr}) && \text{(3.25b)} \\ \leq& \frac{\overline{N^{ub}(0)}}{2^{nr}} && \text{(3.25c)} \\ \leq& O\left(2^{n\left(\frac{w^{ub}(0)}{\ln 2} - r\right)}\right), && \text{(3.25d)} \end{aligned}$$

which converges to 0 as n approaches infinity for all $r > \frac{w^{ub}(0)}{\ln 2}$. Therefore, if $w^{ub}(0) \leq 0$, the nonsystematic (c, d) regular LDGM ensemble essentially suffers no rate reduction with asymptotically high probability. This last inequality is shown to be true in the following theorem.

Theorem 3.2 *For the (c, d) nonsystematic regular LDGM ensembles, $R = d/c$ is the guaranteed rate with asymptotically high probability in the ensemble, i.e., these ensembles suffer no rate reduction.*

Proof: Following the above discussion, it is sufficient to show that $w^{ub}(0) \leq 0$ for these ensembles. Starting form the expression in (3.20) we have for every $b \in [0, 1]$

$$R(1-c)H(b) + \inf_{x>0} \ln \frac{f_+(x,d)}{x^{bd}} \leq (1-d)H(b) + \inf_{x>0} \ln \frac{f_+(x,d)}{x^{bd}}. \tag{3.26}$$

However, the right hand side of this inequality is exactly the AAWD of Gallager's (n, d, d) ensemble as described in Section 2.2 with rate 0. Thus from Fact 2.1, it

follows that

$$w^{ub}(0) = \max_{0\le b\le 1}\left\{R(1-c)H(b) + \inf_{x>0}\ln\frac{f_+(x,d)}{x^{bd}}\right\} \tag{3.27a}$$

$$\le \max_{0\le b\le 1}\left\{(1-d)H(b) + \inf_{x>0}\ln\frac{f_+(x,d)}{x^{bd}}\right\} \tag{3.27b}$$

$$= \max_{b\in\{0,1/2,1\}}\left\{(1-d)H(b) + \inf_{x>0}\ln\frac{f_+(x,d)}{x^{bd}}\right\} \tag{3.27c}$$

$$=0. \tag{3.27d}$$

■

Since the accumulator code maps distinct input words to distinct output words, we have the following corollary.

Corollary 3.1 *The nonsystematic (c,d) regular IRA ensemble suffers no rate reduction, and its guaranteed rate is given by $R = c/d$ with asymptotically high probability.*

3.2.4 Numerical Results

Although the AAWD's given in Theorem 3.1 do not assume close forms, we can still numerically evaluate these expressions to acquire some intuition and insight on the performance of the LDGM and IRA ensembles. Fig. 3.2 depicts the AAWD of the nonsystematic (10,5) regular LDGM and IRA ensembles, and compares them with those of the (7,14) regular LDPC ensemble given in [56] and the rate-1/2 random ensemble. As can be seen in the figure, the IRA ensemble has a more concentrated AAWD than the LDGM ensemble. This demonstrates how the rate-1 accumulator code helps eliminate low weight codewords in the nonsystematic LDGM codes. Moreover, this figure shows that the AAWD of the nonsystematic (10,5) regular IRA ensemble well approximates that of the random ensemble with the same rate for positive values of the AAWD.

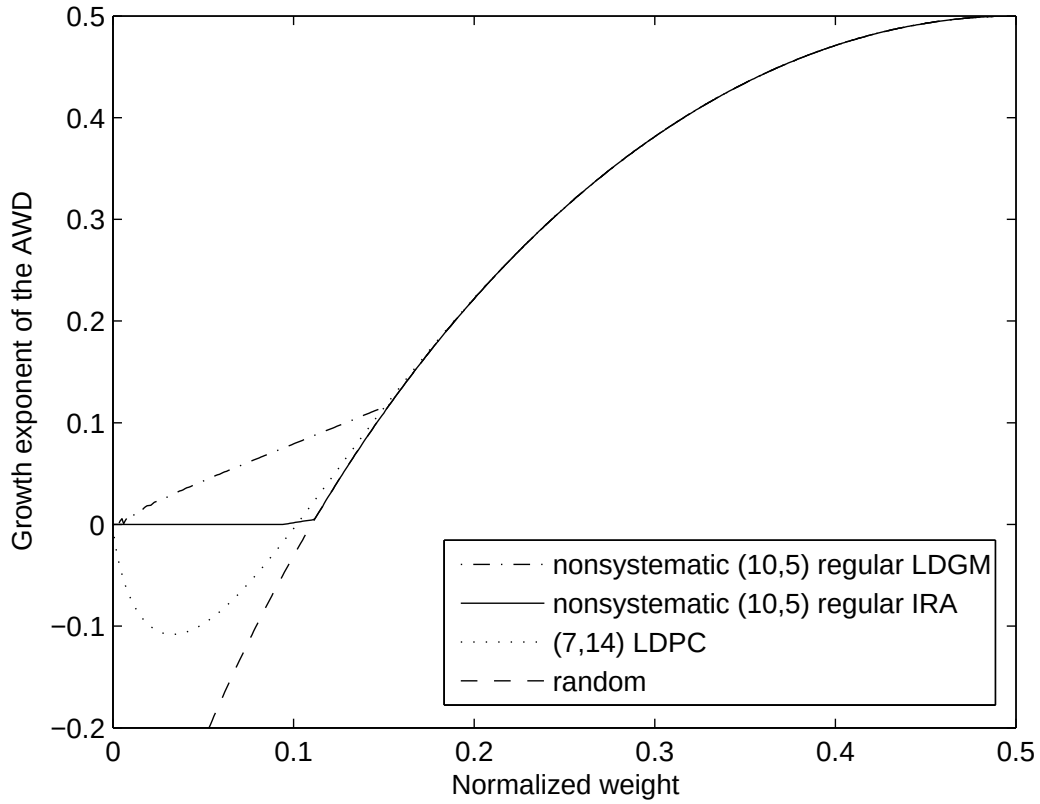


Figure 3.2: The AAWD of the nonsystematic (10,5) regular LDGM ensemble, nonsystematic (10,5) regular RA ensemble, (7,14) regular LDPC ensemble, and the random ensemble. All of them have rate 1/2, and the logarithm is to the base 2.

A similar comparison is shown in Fig. 3.3. The effect of the accumulator code is also evident for both the systematic (12,12) and the nonsystematic (10,5) regular IRA ensembles. Also can be seen in this figure is that the AAWD of the nonsystematic IRA ensemble is better than that of the systematic one with the same graphical complexity. This finding is consistent with the results of [18] for the BEC.

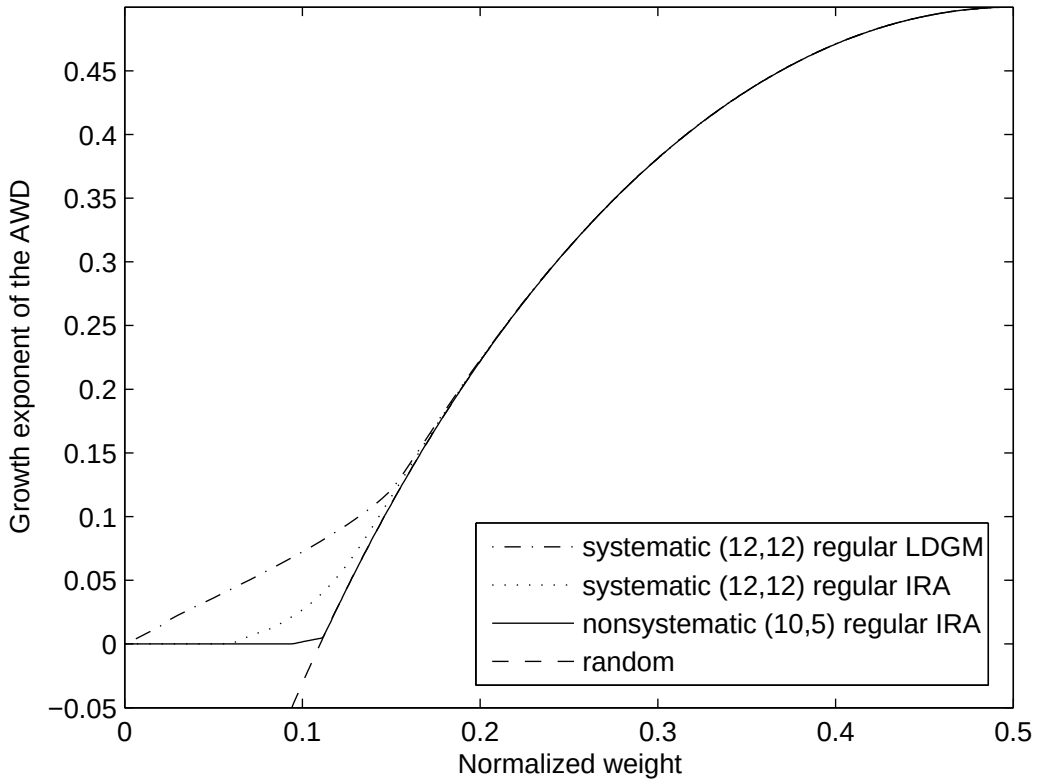


Figure 3.3: The AAWD of the systematic (12,12) regular LDGM ensemble, systematic (12,12) regular RA ensemble, nonsystematic (10,5) regular RA ensemble, and the random ensemble. All of them have rate 1/2, and the logarithm is to the base 2.

Motivated by the above numerical examples, we now focus on the nonsystematic regular IRA ensembles and investigate their AAWD's with different check node degrees and the same rate. Fig. 3.4 shows that the AAWD of the rate 1/2 nonsystematic regular IRA ensembles approaches that of the random code ensemble (for positive values of the AAWD) with increasing check node degrees. In particular, the AAWD of the nonsystematic (12,6) regular IRA ensemble almost coincides with

that of the random ensemble for all growth exponent values greater than 0 with a moderate check node degree equal to $6 + 2 = 8$. Compared with LDPC ensembles, which are proved in [27] to have a random-ensemble-like AAWD only at the limit when the check node degree goes to infinity, IRA ensembles appear to have a great potential of achieving capacity with bounded (small) check node degrees.

However, it is proved in [57] that if a code can be encoded in linear time using sub-linear memory (IRA codes fall in this category) then this code can not have minimum distance growing linearly with n. This result explains why the AAWD of IRA ensembles is always nonnegative as can be seen in the figures, and imposes a difficulty on proving IRA codes to be capacity-achieving on MBIOS channels even with ML decoding. For instance, the ML bound in [48] cannot be used in the same way used to prove that regular LDPC codes can achieve capacity on MBIOS channels with ML decoding [27]. In the next section, we introduce a new family of codes that circumvent this problem and are proved to be capacity-achieving on MBIOS channels with ML decoding.

Table 3.1: Comparison of $(\frac{E_b}{N_0})^*$ as given in (3.28) for several ensembles with rate 1/2.

Ensemble	$\overline{e}$	$(\frac{E_b}{N_0})^*(dB)$
nonsystematic (4,2) regular IRA	8	0.308
Systematic (6,6) regular IRA	8	0.444
(4,8) regular LDPC	8	0.426
nonsystematic (6,3) regular IRA	10	0.308
Systematic (8,8) regular IRA	10	0.343
(5,10) regular LDPC	10	0.341
nonsystematic (8,4) regular IRA	12	0.308
Systematic (10,10) regular IRA	12	0.318
(6,12) regular LDPC	12	0.318
nonsystematic (10,5) regular IRA	14	0.308
Systematic (12,12) regular IRA	14	0.311
(7,14) regular LDPC	14	0.311
Shannon limit		0.184

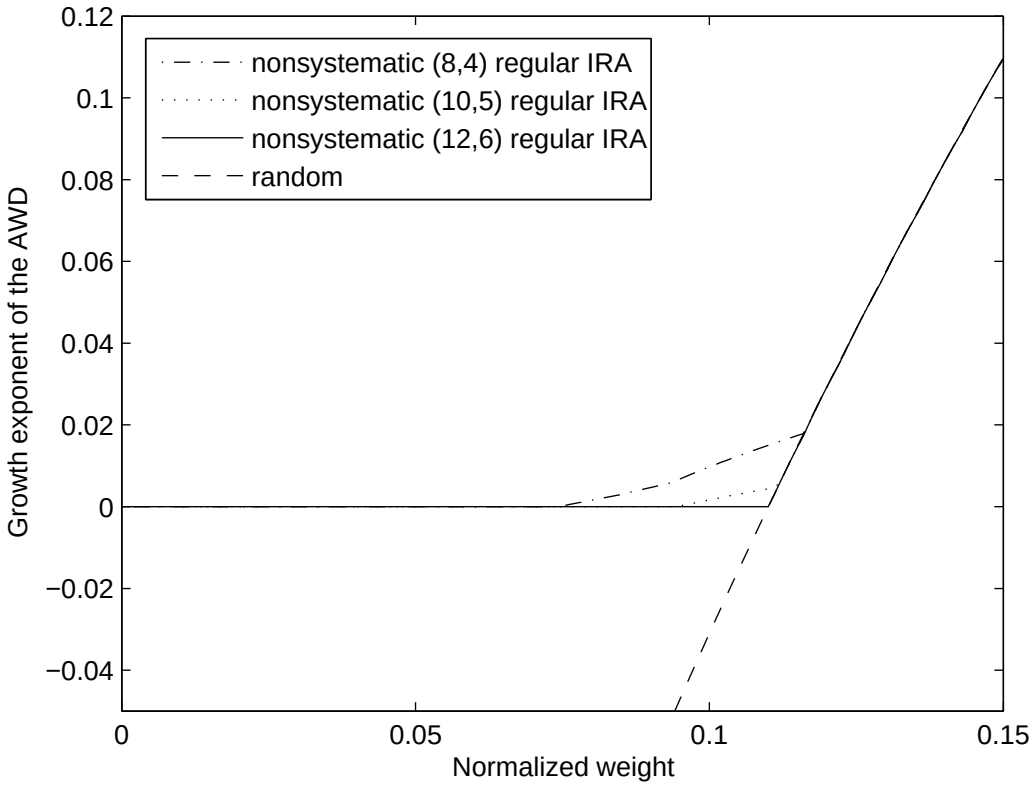


Figure 3.4: Comparison for the AAWD of nonsystematic regular RA ensembles with different right degrees. All of them have rate 1/2, and the logarithm is to the base 2.

To further investigate how regular IRA ensembles perform on the BIAWGN channel with ML decoding, we invoke Divsalar's bound [28] on the minimum bit signal to noise ratio (SNR) $(\frac{E_b}{N_0})^*$ required for reliable communication as follows

$$(\frac{E_b}{N_0})^* \leq \frac{1}{R} \max_{0 \leq a \leq 1} \left\{ \frac{(1 - e^{-2w(a)})(1 - a)}{2a} \right\}. \tag{3.28}$$

This bound is shown in [28] to have the same error exponent as the tangential sphere bound of Poltyrev [47], which often happens to be the tightest reported upper bound for block codes transmitted over the BIAWGN channel (see [58]). The results for rate-1/2 nonsystematic regular IRA, systematic regular IRA and regular LDPC ensembles are summarized in Table 3.1. The comparison is based on the same graphical complexity, denoted by $\overline{e}$ in the table. As can be seen, all nonsystematic regular IRA ensembles with a check node degree greater than 3 yield the same performance bound, which is better than that of their corresponding systematic regular IRA and regular LDPC ensembles, and is only 0.124 dB away from the Shannon limit. This result seems to suggest that nonsystematic regular IRA codes with small check node degrees can come very close to the BIAWGN channel capacity.

3.3 LDPC-GM Codes

In this section, we introduce the LDPC-GM codes, which are serially concatenated codes with an outer LDPC code and an inner LDGM code. In particular, we show that if the outer code is Gallager's LDPC code and the inner code is a rate-1 regular LDGM code, then this LDPC-GM ensemble can achieve capacity on MBIOS channels using ML decoding with bounded graphical complexity.

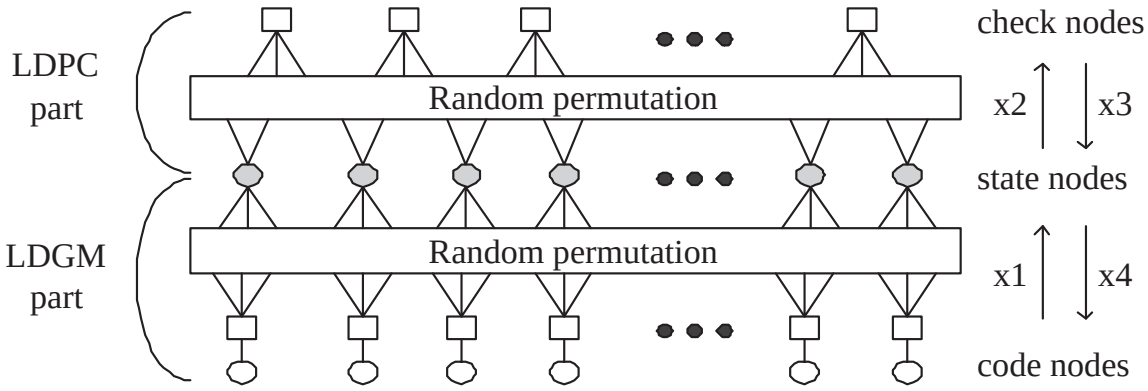


Figure 3.5: The factor graph of the LDPC-GM codes

3.3.1 Concatenation of LDPC and Rate-1 LDGM Codes

Consider the concatenation of an outer Gallager's (n, j, k_1) LDPC code with guaranteed rate $R_o = 1 - j/k_1$ as described in Section 2.2 (assuming k_1 is even as well) and an inner rate-1 (k_2, k_2) regular LDGM code as shown in Fig. 3.5. For simplicity, we assume that $k = k_1 = k_2$ throughout this chapter. If we ignore the possibility that different LDPC codewords can become the same codeword after further encoded by the inner LDGM code and just overcount them, then due to the randomness of the LDPC code construction and from (3.11), the AWD $\overline{N_c(l)}$ of this LDPC-GM ensemble can be bounded by

$$\overline{N_c(l)} \leq \overline{N_c^{ub}(l)} \triangleq \sum_{s=0}^{n} \frac{\overline{N_o(s)}\,\overline{Z_{s,l}^{(LDGM)}}}{\binom{n}{s}} \tag{3.29a}$$

$$= \binom{n}{l} \sum_{s=\lceil l/k \rceil}^{\lfloor n-l/k \rfloor} \frac{\overline{N_o(s)}}{\binom{kn}{ks}} \mathrm{coef}(f_-(x,k)^l f_+(x,k)^{n-l}, x^{ks}), \tag{3.29b}$$

where the change of the range of the summation in the last equality is due to the fact that $\mathrm{coef}(f_-(x,k)^l f_+(x,k)^{n-l}, x^{ks}) = 0$ for $s < \lceil l/k \rceil$ and $s > \lfloor n - l/k \rfloor$. To calculate

the asymptotic growth rate of $\overline{N_c^{ub}(l)}$, we employ (3.18a) and (2.12) to get

$$w_c(a) \triangleq \lim_{n\to\infty} \frac{1}{n} \ln \overline{N_c(an)} \tag{3.30a}$$

$$\leq \lim_{n\to\infty} \frac{1}{n} \ln \overline{N_c^{ub}(an)} \tag{3.30b}$$

$$= H(a) + \max_{\frac{a}{k} \leq b \leq 1-\frac{a}{k}} w_o(b) - kH(b) + \inf_{x>0} \ln \frac{f_-(x,k)^a f_+(x,k)^{1-a}}{x^{bk}} \tag{3.30c}$$

$$\overset{(a)}{\leq} H(a) + \max_{\frac{a}{k} \leq b \leq 1-\frac{a}{k}} w_o(b) + a \ln[1-(1-2b)^k] + (1-a)\ln[1+(1-2b)^k] - \ln 2 \tag{3.30d}$$

$$\triangleq w_c^{ub}(a), \tag{3.30e}$$

where inequality (a) follows by substituting $x = \frac{b}{1-b}$ in the infimum expression. Since Theorem 3.2 shows that regular LDGM ensembles are free of rate reduction with asymptotically high probability, we conclude that the guaranteed rate R of this LDPC-GM ensemble is R_o with asymptotically high probability.

3.3.2 Analysis of the LDPC-GM Codes

In this section, we will first characterize $w_c^{ub}(a)$, and then use the derived results to prove that LDPC-GM codes can be capacity-achieving on MBIOS channels using ML decoding with bounded graphical complexity. Although $w_c^{ub}(a)$ is not symmetric about $a = 1/2$, the following lemma shows that we can focus on analyzing $w_c^{ub}(a)$ for $a \in [0, 1/2]$ and bound $w_c^{ub}(a)$ by $w_c^{ub}(1-a)$ for $a \in [1/2, 1]$.

Lemma 3.1 *$w_c^{ub}(a) \leq w_c^{ub}(1-a)$ for all $a \in [1/2, 1]$.*

Proof: Since

$$\ln[1-(1-2b)^k] \leq 0 \leq \ln[1+(1-2b)^k], \quad \forall b \in [0,1], \tag{3.31}$$

and $1-a \leq a$ for all $a \in [1/2, 1]$, it follows from (3.30) that for all $a \in [1/2, 1]$, we have

$$w_c^{ub}(a) = H(a) + \max_{\frac{a}{k} \leq b \leq 1-\frac{a}{k}} w_o(b) + a \ln[1-(1-2b)^k] + (1-a)\ln[1+(1-2b)^k] - \ln 2 \tag{3.32a}$$

$$\leq H(1-a) + \max_{\frac{1-a}{k} \leq b \leq 1-\frac{1-a}{k}} w_o(b) + (1-a)\ln[1-(1-2b)^k] + a\ln[1+(1-2b)^k] - \ln 2 \tag{3.32b}$$

$$= w_c^{ub}(1-a). \tag{3.32c}$$

■

In the next theorem, we prove that given any R_1 in [0, 1], the positive part of $w_c^{ub}(a)$ can be upper bounded by the AAWD of the random ensemble for any rate $R \leq R_1$ if k is sufficiently large. In this case, we also prove that $\overline{N_c(l)}$ decreases at least polynomially with n in the negative part of $w_c^{ub}(a)$ when $j \geq 3$.

Theorem 3.3 *For any $R_1 \in [0, 1]$, there exists an integer $M < \infty$ such that for all $k > M$ and for all LDPC-GM codes with design rate $R \in [0, R_1]$, there exists a $\delta' < H^{-1}((1-R)\ln 2)$ such that the following two statements are true.*

1.
$$w_c^{ub}(a) \begin{cases} \leq 0 & \text{if } a = 0, \\ < 0 & \text{if } a \in (0, \delta'], \\ \leq H(a) - (1-R)\ln 2 & \text{if } a \in (\delta', 1/2]. \end{cases} \tag{3.33}$$

2. $\overline{N_c^{ub}(l)} = O(n^{-j+2})$ *for all* $l \in (0, \delta' n] \cup [n - \delta' n, n]$.

Proof: See Appendix B.1. ■

One important point in the above theorem is that, M only depends on R_1, and does not vary with the design rate R. Therefore, given any MBIOS channel with capacity C, if we let $R_1 = C$, then a fixed and finite $k > M$ (e.g., $k = M + 1$) depending only on C is sufficient for all LDPC-GM codes of design rate $R \in [0, C]$ to satisfy the two statements of Theorem 3.3. On the contrary, if M were also dependent on R (as proved in [27] for the case of regular LDPC codes), then M (and thus the graphical complexity) could approach infinity as R approaches the capacity C. Indeed, this is the essence of our bounded graphical complexity result as will be proved in Theorem 3.4.

If we let d_{min} and $N_c(l)$ be the random variables denoting the minimum distance and number of codewords of weight l, respectively, of a randomly drawn code from the LDPC-GM ensemble, and if we denote by $\delta_{GV} = H^{-1}((1-R)\ln 2)$ the normalized Gilbert-Varshamov distance, then from Markov's inequality, we have for all $\epsilon > 0$,

$$P(d_{min} \leq (\delta_{GV} - \epsilon)n) = P\left(\sum_{l \in (0, (\delta_{GV}-\epsilon)n]} N_c(l) \geq 1\right) \tag{3.34a}$$

$$\leq \sum_{l \in (0, (\delta_{GV}-\epsilon)n]} \overline{N_c(l)} \tag{3.34b}$$

$$\leq n \max_{l \in (0, \delta' n]} \overline{N_c^{ub}(l)} + n \exp\left\{ n \sup_{a \in (\delta', \delta_{GV}-\epsilon]} w_c^{ub}(a) + o(n) \right\} \tag{3.34c}$$

$$= O(n^{-j+3}), \tag{3.34d}$$

where the last equality follows from the facts proved in Theorem 3.3. In particular, for the first term in (3.34c) we have $\overline{N_c^{ub}(l)} = O(n^{-j+2})$ for all $l \in (0, \delta' n]$ and for the

second term we have

$$w_c^{ub}(a) < 0 \text{ for all } a \in [\delta', \delta_{GV} - \epsilon] \tag{3.35a}$$

$$\Rightarrow n \exp\left\{ n \sup_{a \in (\delta', \delta_{GV} - \epsilon]} w_c^{ub}(a) + o(n) \right\} \text{ decreases exponentially for large } n. \tag{3.35b}$$

Equation (3.34) implies that $P(d_{min} > (\delta_{GV} - \epsilon)n)$ approaches 1 asymptotically as n approaches infinity when $j \geq 4$. Since ϵ can be arbitrarily small, we have proved the following corollary.

Corollary 3.2 *For any $R_1 \in [0, 1]$, there exists an integer $M < \infty$ such that for all $k > M$, and for all $j \geq 4$, all LDPC-GM codes with design rate $R \in [0, R_1]$ have a normalized minimum distance, which is arbitrarily close to the Gilbert-Varshamov bound with asymptotically high probability.*

In Fig. 3.6, we compare the AAWD of the LDPC, the LDPC-GM and the random ensemble with $R = 0.5$ and $k = 8$. It is evident that the rate-1 LDGM inner code helps eliminate high weight codewords in the outer LDPC code. As a trade-off, the growth rate of some low weight codewords increases slightly. However, as long as the growth rate of the low weight codewords remains negative, the number of low weight codewords still becomes vanishingly small as n goes to infinity.

We are now ready to state the main theorem of this section, which shows that given any MBIOS channel, there always exists a finite value M such that the LDPC-GM ensemble with $k > M$ is capacity-achieving.

Theorem 3.4 *Given any MBIOS channel with capacity C, there exists an integer $M < \infty$ such that the average block error probability P_B of the LDPC-GM ensembles with $k > M$, $j \geq 4$ and rate $R < C$ is vanishingly small when ML decoding is used.*

Proof: See Appendix B.2. ■

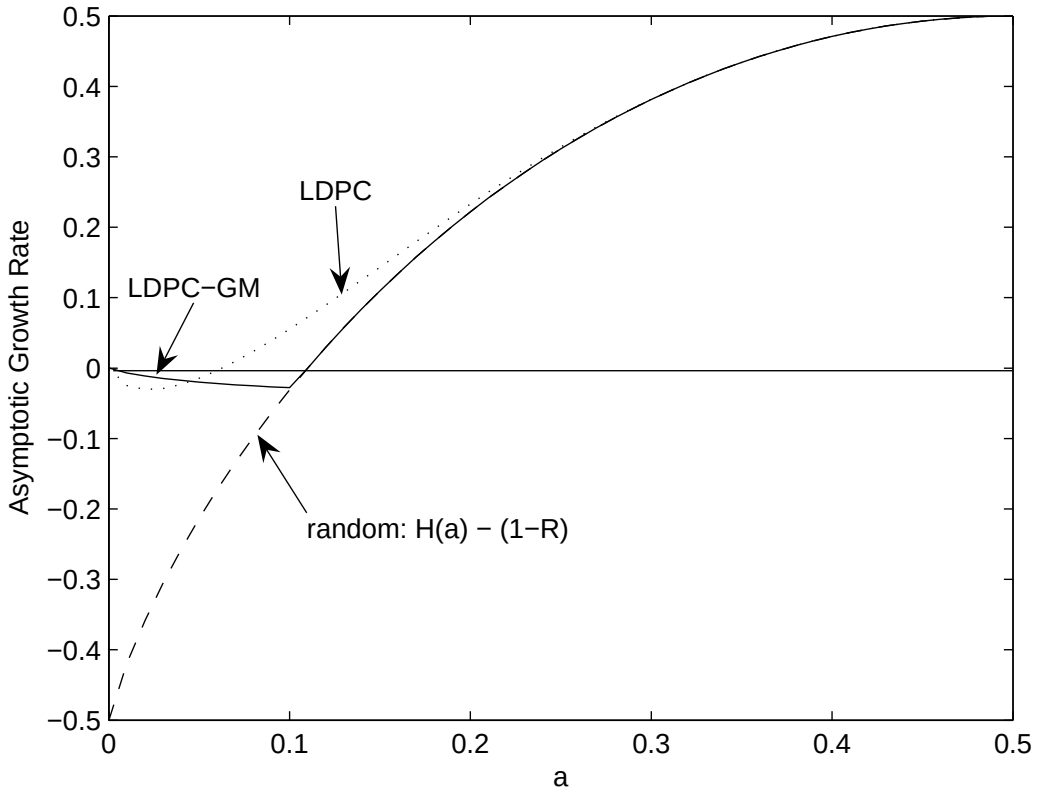


Figure 3.6: Comparison of $w_o(a)$, $w_c(a)$ and $H(a) - (1 - R)$ with $R = 0.5$ and $k = 8$. The logarithm is to the base 2 in this figure.

As can be seen in Fig. 3.5, the graphical complexity Δ of these LDPC-GM codes can be evaluated as follows

$$\Delta = \frac{n(j+k)+n}{Rn} = \frac{(2-R)k+1}{R}. \tag{3.36}$$

Since Theorem 3.4 guarantees that k does not approach infinity, we can deduce that these LDPC-GM codes with any rate $R \in (0,1)$ can be capacity-achieving on any MBIOS channel with bounded graphical complexity.

3.4 Density Evolution for LDPC-GM Codes on the BEC

Although the aforementioned LDPC-GM ensembles have finite graphical complexity, the decoding complexity under ML decoding is still exponential. In this section, we show that by allowing the outer LDPC and inner LDGM codes to be more generally irregular, there exist capacity-achieving LDPC-GM ensembles on any BEC under BP decoding with bounded decoding complexity per information bit. Although this is not a proof that the same might be true for MBIOS channels, it is a good indication of the potential of the LDPC-GM codes.

Consider the concatenation of a (λ, ρ) irregular LDPC code and a $(2,2)$ regular LDGM code, where λ and ρ are the standard variable and check node degree distributions, respectively, from the edge perspective as defined in [59]. Note that this LDPC-GM ensemble has guaranteed rate

$$R = 1 - \frac{\int_0^1 \rho(t)dt}{\int_0^1 \lambda(t)dt}. \tag{3.37}$$

Let q be the channel erasure probability, and let x_1, x_2, x_3 and x_4 be the probabilities

of erasure on edges from check to variable (LDGM), variable to check (LDPC), check to variable (LDPC) and variable to check (LDGM) nodes, respectively, as shown in Fig. 3.5. Then, assuming we are operating at some fixed point, we have the following density evolution equations[4].

$$x_1 = 1 - (1-q)(1-x_4) \tag{3.38a}$$

$$x_2 = x_1^2 \lambda(x_3) \tag{3.38b}$$

$$x_3 = 1 - \rho(1-x_2) \tag{3.38c}$$

$$x_4 = x_1 \tilde{\lambda}(x_3), \tag{3.38d}$$

where $\tilde{\lambda}(x) = \sum_{i=1}^{\infty} \tilde{\lambda}_i x^i$ and

$$\tilde{\lambda}_i = \frac{\lambda_i / i}{\int_0^1 \lambda(t)dt}, \tag{3.39}$$

denoting the fraction of variable nodes in the LDPC code with degree i. Equivalently, we have

$$\tilde{\lambda}(x) = \frac{\int_0^x \lambda(t)dt}{\int_0^1 \lambda(t)dt}. \tag{3.40}$$

Note that equations (3.38) are also the density evolution equations for the serially concatenated codes with an outer LDPC code and an inner differentiator code[5].

[4]The density evolution method can be applied here since LDPC-GM codes are instances of multi-edge type LDPC codes [16].

[5]A differentiator code with input a_k and output b_k is defined by the input-output relation $b_k = a_{k-1} + a_k$

Solving these equations for x_3, we have

$$\begin{aligned} x_3 =& 1-\rho(1-x_2) && \text{(3.41a)}\\ =& 1-\rho(1-x_1^2\lambda(x_3)) && \text{(3.41b)}\\ =& 1-\rho\left(1-\left[\frac{q}{1-(1-q)\tilde{\lambda}(x_3)}\right]^2\lambda(x_3)\right). && \text{(3.41c)} \end{aligned}$$

where the last equality follows from the following:

$$\begin{aligned} & x_4 = x_1\tilde{\lambda}(x_3) = [1-(1-q)(1-x_4)]\tilde{\lambda}(x_3) && \text{(3.42a)}\\ \Rightarrow & x_4\left[1-(1-q)\tilde{\lambda}(x_3)\right] = q\tilde{\lambda}(x_3) && \text{(3.42b)}\\ \Rightarrow & x_4 = \frac{q\tilde{\lambda}(x_3)}{1-(1-q)\tilde{\lambda}(x_3)} && \text{(3.42c)}\\ \Rightarrow & 1-x_4 = \frac{1-\tilde{\lambda}(x_3)}{1-(1-q)\tilde{\lambda}(x_3)} && \text{(3.42d)}\\ \Rightarrow & x_1 = 1-(1-q)(1-x_4) = \frac{q}{1-(1-q)\tilde{\lambda}(x_3)}. && \text{(3.42e)} \end{aligned}$$

If (3.41) has no solution in $(0,1]$, then x_3 must converge to 0, and thus x_4 must converge to 0 as the number of iterations approaches infinity. Therefore, if we have

$$1-\rho\left(1-\left[\frac{q}{1-(1-q)\tilde{\lambda}(x_3)}\right]^2\lambda(x_3)\right) < x_3, \quad \forall x_3 \in (0,1] \qquad \text{(3.43)}$$

then the BP decoding is successful. Note that, (3.41) is essentially the same as equation (6) in [18] except for the following changes: $x_0 \to 1-x_3$, $p \to 1-q$, $\lambda(\cdot) \to \rho(\cdot)$, $\rho(\cdot) \to \lambda(\cdot)$, and $R(\cdot) \to \tilde{\lambda}(\cdot)$. More generally, (3.41) is an instance of the symmetry introduced in [52]. So, in the following, we will use the results proved in [18] to show two particular degree distribution pairs are capacity-achieving under BP decoding.

Theorem 3.5 (Check-regular ensemble) *Let*

$$\lambda(x) = \frac{1-(1-x)^{\frac{1}{k-1}}}{\left[1-(1-q)\left(1-kx+(k-1)\left[1-(1-x)^{\frac{k}{k-1}}\right]\right)\right]^2} \tag{3.44}$$

$$\rho(x) = x^{k-1}. \tag{3.45}$$

Then for $k = 3$ and $q \in [\frac{12}{13}, 1)$, $\lambda(x)$ has only non-negative coefficients. Moreover, for any $\epsilon \in (0,1)$, let $M(\epsilon)$ be the smallest positive integer such that[6]

$$\sum_{i=M(\epsilon)+1}^{\infty} \frac{\lambda_i}{i} < \frac{\epsilon(1-q)}{qk}, \tag{3.46}$$

and let $\lambda_\epsilon(x)$ be the truncated degree distribution of $\lambda(x)$ by treating all variable nodes with degree greater than $M(\epsilon)$ as pilot bits. Then the degree distribution pair (λ_ϵ, ρ) achieves a fraction $1-\epsilon$ of the channel capacity with vanishing bit error probability under BP decoding.

Proof: See Appendix B.3.1. ■

The decoding complexity per information bit of this check-regular ensemble can be calculated as follows

$$\Delta < \frac{knq + 2n + n}{(1-q)(1-\epsilon)n} = \frac{qk+3}{(1-q)(1-\epsilon)}, \tag{3.47}$$

which approaches the bounded value $\frac{qk+3}{1-q}$ as ϵ approaches 0.

[6] $M(\epsilon)$ exists for all $\epsilon \in (0,1)$ since $\sum_{i=1}^{\infty} \frac{\lambda_i}{i} = \int_0^1 \lambda(t)dt = \frac{1}{qk}$, which means $\sum_{i=M(\epsilon)+1}^{\infty} \frac{\lambda_i}{i}$ can be made arbitrarily close to 0 by increasing $M(\epsilon)$.

Theorem 3.6 (Variable-regular ensemble) *Let*

$$\lambda(x) = x^2 \tag{3.48}$$

$$\rho(x) = 1 + \frac{2(1-q)(1-x)^2 \sin\left(\frac{1}{3}\arcsin\left(\sqrt{-\frac{27(1-q)(1-x)^{\frac{3}{2}}}{4q^3}}\right)\right)}{\sqrt{3}q^4\left[-\frac{(1-q)(1-x)^{\frac{3}{2}}}{q^3}\right]^{\frac{3}{2}}}. \tag{3.49}$$

Then for $q \in [0.05, 1]$, $\rho(x)$ *has only non-negative coefficients. Moreover, for any* $\epsilon \in (0,1)$, *let* $M(\epsilon)$ *be the smallest positive integer such that*[7]

$$\sum_{i=M(\epsilon)+1}^{\infty} \rho_i < \frac{\epsilon(1-q)}{3}, \tag{3.50}$$

and let

$$\rho_\epsilon(x) \triangleq \left(1 - \sum_{i=1}^{M(\epsilon)} \rho_i\right) + \sum_{i=1}^{M(\epsilon)} \rho_i x^{i-1} \tag{3.51}$$

be the truncated degree distribution of $\rho(x)$. *Then the degree distribution pair* (λ, ρ_ϵ) *achieves a fraction* $1-\epsilon$ *of the channel capacity with vanishing bit error probability under BP decoding.*

Proof: See Appendix B.3.2. ■

The decoding complexity per information bit of this variable-regular ensemble can be calculated as follows

$$\Delta < \frac{3n + 2n + n}{(1-q)(1-\epsilon)n} = \frac{6}{(1-q)(1-\epsilon)}, \tag{3.52}$$

which approaches the bounded value $\frac{6}{1-q}$ as ϵ approaches 0.

[7] $M(\epsilon)$ exists for all $\epsilon \in (0,1)$ since $\sum_{i=1}^{\infty} \rho_i = 1$, which means $\sum_{i=M(\epsilon)+1}^{\infty} \rho_i$ can be made arbitrarily close to 0 by increasing $M(\epsilon)$.

One drawback of these capacity-achieving degree distribution pairs is that they are not guaranteed to be valid, i.e., with only nonnegative coefficients, for all $q \in (0, 1)$. Thus, it is not clear if there exist capacity-achieving check-regular ensembles with bounded complexity for the BEC with erasure probability $q < 12/13$ (respectively, variable-regular ensembles for the BEC with erasure probability $q < 0.05$). However, since this is true for q in the vicinity of 1 (which is not true for the capacity-achieving IRA codes in [18]), it is possible to construct capacity-achieving ensembles with bounded complexity for any q by considering punctured LDPC-GM codes. The idea is to construct a low-rate capacity-achieving code for a bad (i.e., q close to 1) BEC channel, and then use puncturing to increase its rate. In [60], it is shown that random puncturing results in no performance loss on the gap to capacity for codes on the BEC. It follows that puncturing can be used to increase the rate of LDPC-GM codes without affecting their capacity-achievability, a fact that was also observed by Pfister and Sason [52]. Furthermore, since a punctured LDPC-GM ensemble can also be viewed as another unpunctured LDPC-GM ensemble with inner irregular LDGM code (which is no longer rate-1 in general), we have the following theorem.

Theorem 3.7 *Let (λ, ρ) be a degree distribution pair implied by Theorem 3.5 or Theorem 3.6 for some given ϵ and q'. Consider the LDPC-GM ensemble, whose outer LDPC code has degree distribution pair (λ, ρ), and inner LDGM code has degree distribution pair (f, g). Then for any given $p \in [0, q']$, if*

$$f(x) = x(1 - p) + p \tag{3.53a}$$

$$g(x) = x, \tag{3.53b}$$

then this LDPC-GM ensemble achieves a fraction of $1 - \epsilon$ of the channel capacity on the BEC with erasure probability $q \triangleq \frac{q'-p}{1-p}$ under BP decoding.

Proof: See Appendix B.3.3. ■

According to this theorem, given any capacity-achieving degree distribution pair (λ, ρ) for some erasure probability q', we can generate capacity-achieving LDPC-GM ensembles for all erasure probabilities $q \in [0, q']$ by adjusting p. Since q' can be arbitrarily close to 1, and the maximum degrees of f and g are bounded for all p, this construction can produce capacity-achieving LDPC-GM ensembles for all BECs with bounded decoding complexity per information bit.

3.5 Conclusion

In this chapter, we first studied some fundamental properties of IRA codes, and then introduced LDPC-GM codes, which were proved to be capacity-achieving on the BEC and MBIOS channels using BP and ML decoding, respectively, with bounded graphical complexity. These LDPC-GM codes are the first reported ensembles that achieve capacity on MBIOS channels with bounded graphical complexity.

For the IRA codes, we derived their AIPWE by viewing an IRA code as a serially concatenated code with an outer LDGM code and an inner accumulator code. The resulting AIPWE was then used to derive the AWD for systematic and nonsystematic versions of IRA codes, and their asymptotic growth rate was also calculated. In the process, we also derived the AAWD of the systematic and nonsystematic LDGM codes, and proved that nonsystematic regular LDGM codes are free of rate reduction with asymptotically high probability. By numerically evaluating the AAWD of the ensembles, we concluded that: (a) the accumulator code plays an important role in eliminating low weight codewords for IRA ensembles; (b) the nonsystematic regular IRA ensembles have more concentrated AAWD's than their corresponding systematic ones with the same graphical complexity; (c) the nonsystematic regular IRA ensembles with moderate check node degrees have AAWD's very close to

that of the random ensemble for all growth rate values greater than zero. The bit SNR thresholds on the BIAWGN channel based on Divsalar's bound were obtained to show that nonsystematic regular IRA ensembles with small check node degrees have a better guaranteed ML performance than the corresponding systematic regular IRA and regular LDPC ensembles with the same graphical complexity, which is only 0.124 dB away from the Shannon limit. However, although these promising results made nonsystematic IRA ensembles strong candidates for capacity-achieving codes on noisy channels with bounded graphical complexity, the fact that they do not have linearly increasing minimum distance as proved in [57] prevented this statement from being rigorously proved.

Motivated by this reason, LDPC-GM codes, i.e., concatenated codes with an outer LDPC code and an inner LDGM code, were introduced. In the case that the outer code is a Gallager's (n, j, k) LDPC code and the inner code is a rate-1 (k, k) regular LDGM code, we proved that for any desired range of rates R, there always exists an integer $M < \infty$ such that if $k > M$, then the resulting AAWD of the LDPC-GM codes has a positive part, which can be upper bounded by the AAWD of the random ensemble, and a negative part, where the number of codewords vanishes at least polynomially in n when $j \geq 4$. This result was attributed to the presence of the rate-1 LDGM encoder, which helps eliminate high weight codewords while maintaining a vanishingly small amount of low weight codewords in the LDPC code. The implication of the above statement is that these codes achieve the Gilbert-Varshamov bound with asymptotically high probability. Furthermore, after applying the ML performance bound given in [48] to these LDPC-GM codes, we proved that they can achieve capacity on any MBIOS channel using ML decoding. Since all these results hinged on the only condition that k is greater than some finite number, we have proved that these LDPC-GM codes are capacity-achieving codes with bounded

graphical complexity on any MBIOS channel. Finally, if the outer LDPC code is allowed to be irregular, then invoking the density evolution method, we showed two particular ensembles of the LDPC-GM codes to be capacity-achieving on the BEC under BP decoding with bounded decoding complexity. Moreover, extensions valid for all erasure probabilities of the BEC using inner irregular LDGM codes were also presented.

These favorable results could suggest a high potential for the LDPC-GM codes to achieve capacity on MBIOS channels with bounded decoding complexity per iteration under BP decoding. This remains an open problem mainly due to the fact that the only available method for studying the performance of turbo-like codes under BP is density evolution, which becomes ineffective as an analytical tool in MBIOS channels. Even if this problem could be solved another open problem arises: since the required number of iterations for successful iterative decoding for the LDPC-GM codes as a function of the gap to capacity remains unknown (a fact that is true also for all other known capacity-achieving codes), it is not clear whether bounded graphical complexity property implies bounded decoding complexity using iterative decoding.

CHAPTER 4

Iterative Decoding Performance Bounds for LDPC Codes

4.1 Introduction

Since the main difficulty in using the density evolution (DE) method for the asymptotic performance analysis of codes with iterative decoding on memoryless binary-input output-symmetric (MBIOS) channels is that the evolved densities in general require an infinite-dimensional description, one way to circumvent this problem is to track the evolution of the densities projected to some finite-dimensional space. Specifically proposed approaches in this category include the Gaussian approximation [43], the extrinsic information transfer (EXIT) chart [61], and the generalized EXIT (GEXIT) [62] chart methods. Unfortunately, since the EXIT and GEXIT chart methods still require numerical calculations, and the Gaussian approximation does not imply any upper or lower bounds on the exact performance, these results still can not be used to analytically bound the code performance.

In [25], the authors proposed to map the evolved densities to Bhattacharyya parameters, which are further used to bound the bit error probability of the low-

density parity-check (LDPC) codes under sum-product (SP) decoding [8] on MBIOS channels. It turns out that an upper bound on the Bhattacharyya parameters can be obtained by a recursion involving only one-dimensional real numbers, so the whole result can be used to determine a guaranteed decoding capability for the LDPC codes. In this chapter, we improve this result of [25] by showing that the same recursively determined upper bound on the bit error probability of the LDPC codes not only holds for the SP decoding, but also holds for the min-sum (MS) decoding [8]. This result is attained by upper bounding the probability of error of the root bit of a tree code by a sequence error probability of a subcode of the tree code, and then using the union bound. Therefore, the whole proof does not involve the Bhattacharyya parameters.

Then, we turn our attention to SP decoding, and derive recursive lower and upper bounds on the probability of bit error and the Bhattacharyya parameter, respectively, after each iteration. The recursive upper bound on the Bhattacharyya parameter is tighter than the one given in [25], and is also found independently by Jin and Richardson in [30]. Both derived lower and upper bounds become exact and recover the one-dimensional DE equation on the binary erasure channel (BEC). Exploiting the resemblance of our recursions to the DE equation on the BEC, we prove that LDPC codes under SP decoding on the BEC always have the best asymptotic performance among all MBIOS channels with the same uncoded bit error probability, and always have the worst one among all MBIOS channels with the same Bhattacharyya parameter. In other words, the one-dimensional DE equation of LDPC codes on the BEC can be used to bound the SP decoding performance of LDPC codes from below and above on all MBIOS channels. For instance, since an abundance of studies have resulted in LDPC codes working reliably on any given BEC, we can always find codes working reliably on any given MBIOS channel according to this result. Note

also that, due the nature of the proofs of the main lemmas, this performance connection between the BEC and MBIOS channels is also true for the general family of multi-edge type LDPC codes [16], including the irregular repeat-accumulate (IRA) codes [17, 18] and the low-density parity-check and generator matrix (LDPC-GM) codes introduced in Chapter 3.

The remaining of this chapter is structured as follows. In Section 4.2, we review the preliminary background on Bhattacharyya parameters, MBIOS channels and the asymptotic performance analysis of LDPC codes. Then, we present our asymptotic performance analysis of LDPC codes on MBIOS channels under MS and SP decoding in Section 4.3 and 4.4, respectively. Finally, we conclude this chapter in Section 4.5.

4.2 Preliminaries

Let the random variable Y from the alphabet $\mathcal{Y}$ be an observation of the binary variable x from the alphabet $\mathcal{X} \triangleq \{0, 1\}$. Suppose the statistics of Y are completely characterized by the conditional probability density function[1] (pdf) $f(y|x)$. In the two-hypothesis testing problem of decoding x given a realization y of Y, the Bhattacharyya parameter B is defined as

$$B \triangleq \int_{\mathcal{Y}} \sqrt{f(y|0)f(y|1)}dy, \tag{4.1}$$

and it is shown in [64, Theorem 7.5] that B is an upper bound on the maximum-likelihood (ML) decoding error probability. A property of B following from the

[1]Mixed and discrete random variables can also be accommodated by including the Dirac delta function to probability density functions. See [63, Section 5.3] for a discussion.

Cauchy-Schwarz inequality is that

$$0 \leq B = \int_{\mathcal{Y}} \sqrt{f(y|0)f(y|1)}dy \leq \sqrt{\left(\int_{\mathcal{Y}} f(y|0)dy\right)\left(\int_{\mathcal{Y}} f(y|1)dy\right)} = 1. \tag{4.2}$$

If x and y are the input and output symbols, respectively, of an MBIOS channel, then we may assume that $\mathcal{Y}$ is the set of real numbers $\mathbb{R}$ and further have the symmetry condition:

$$f(y|0) = f(-y|1), \quad \forall y \in \mathbb{R}. \tag{4.3}$$

To analyze the asymptotic average iterative decoding performance of a (λ, ρ) irregular LDPC ensemble when the codeword length goes to infinity, where λ and ρ are the degree distribution pair as defined in (1.1), it is shown in Fact 1.2 that we can as well consider the cycle-free case. In this case, the probability of bit error after l decoding iterations is the probability of decoding error of the root bit on the computation tree of $l + 1$ (variable node) levels whose construction is dictated by the degree distributions (λ, ρ) as in [59]. See Fig. 4.1 for an example of the computation tree. Due to the symmetry condition of MBIOS channels, we can also assume without loss of generality that the all-zero codeword is transmitted. Hence, in the following two sections, we will analyze the asymptotic MS and SP iterative decoding performances of the (λ, ρ) LDPC ensemble on MBIOS channels by considering the corresponding tree codes and assuming the transmission of the all-zero codeword.

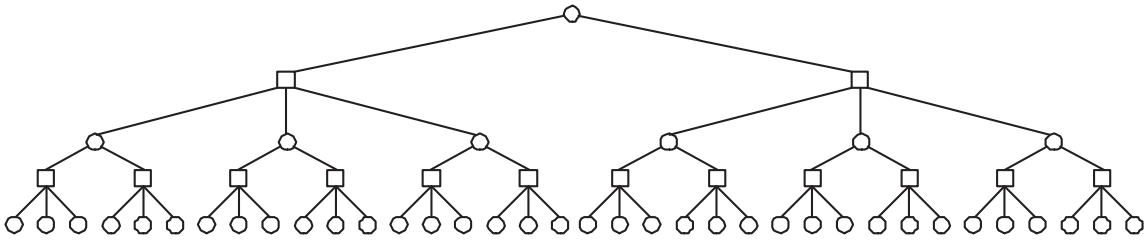


Figure 4.1: Computation tree of the (x^2, x^3) LDPC ensemble of level 3 (two iterations), where circles denote variable nodes and squares denote check nodes.

4.3 Min-Sum Decoding Performance Analysis

Consider an arbitrary binary tree code whose codebook is $\mathcal{C} = \{\mathbf{c}_0 = \mathbf{0}, \mathbf{c}_1, \ldots, \mathbf{c}_M\}$, where $\mathbf{c}_i = (c_{i1}, c_{i2}, \ldots, c_{in})$ is a codeword of length n for all i, and $\mathbf{c}_0$ is the all-zero codeword. Let c_{i1} be the root bit of this tree for all i, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ the transmitted codeword, and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ the received sequence from an MBIOS channel with conditional pdf $f(y|x)$. When MS decoding is performed on this tree, it essentially performs maximum-likelihood sequence detection (MLSqD) on the whole sequence to produce an estimate $\hat{\mathbf{c}} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n)$. Therefore, if we define the decision region for the codeword $\mathbf{c}_i$ as

$$\mathcal{U}_i \triangleq \bigcap_{k \neq i} \mathcal{U}_{ik}, \tag{4.4}$$

where

$$\mathcal{U}_{ik} \triangleq \left\{ \mathbf{y} \in \mathbb{R}^n | \prod_{j=1}^{n} f(y_j|c_{ij}) \geq \prod_{j=1}^{n} f(y_j|c_{kj}) \right\}, \tag{4.5}$$

then the probability of the root bit being in error under MS decoding assuming the transmission of the all-zero codeword $\mathbf{c}_0$ is

$$P_b^{MS} = P(\hat{c}_1 = 1|\mathbf{x} = \mathbf{c}_0) = P\left(\mathbf{y} \in \bigcup_{i, c_{i1}=1} \mathcal{U}_i | \mathbf{x} = \mathbf{c}_0 \right). \tag{4.6}$$

We would like to find a compact representation of the set

$$\mathcal{U} \triangleq \bigcup_{i,c_{i1}=1} \mathcal{U}_i, \tag{4.7}$$

or at least a superset of $\mathcal{U}$ so that we can bound P_b^{MS} from above. Note that such an upper bound is also a valid upper bound on the probability of root bit error P_b^{SP} when the SP decoding is performed on the tree since the SP decoding essentially performs optimal ML decoding for each bit.

Definition 4.1 *Define the subcode $\mathcal{C}_r$ of $\mathcal{C}$ as the set of codewords in $\mathcal{C}$ such that*

(i) each check node with parent variable node equal to 1 has exactly one child variable node equal to 1, and

(ii) each check node with parent variable node equal to 0 has all child variable nodes equal to 0.

Note that in this reduced codebook $\mathcal{C}_r$, the only codeword with root bit equal to 0 is the all-zero codeword. Moreover, as can be seen in Fig. 4.2, the number of codewords with root bit equal to 1 is reduced in $\mathcal{C}_r$. In the following lemma, we show how we can use this reduced codebook $\mathcal{C}_r$ to characterize $\mathcal{U}$.

Lemma 4.1 $\mathcal{U} \subset \bigcup_{i,c_{i1}=1,\mathbf{c}_i\in\mathcal{C}_r} \mathcal{U}_{i0}$.

Proof: Given any $\mathbf{y} \in \mathcal{U} = \bigcup_{i,c_{i1}=1} \mathcal{U}_i$, there exists a k such that $\mathbf{y} \in \mathcal{U}_k$ and $c_{k1} = 1$. To proceed with our proof, we first carry out the following labeling procedure on the codeword $\mathbf{c}_k$.

1. At the initial state, the root bit is labelled as "survived", and all the other variable nodes are unlabelled. Note that in this labeling procedure, the "survived" variable nodes will always have value 1 in $\mathbf{c}_k$. We first consider the check nodes at the topmost level of the tree.

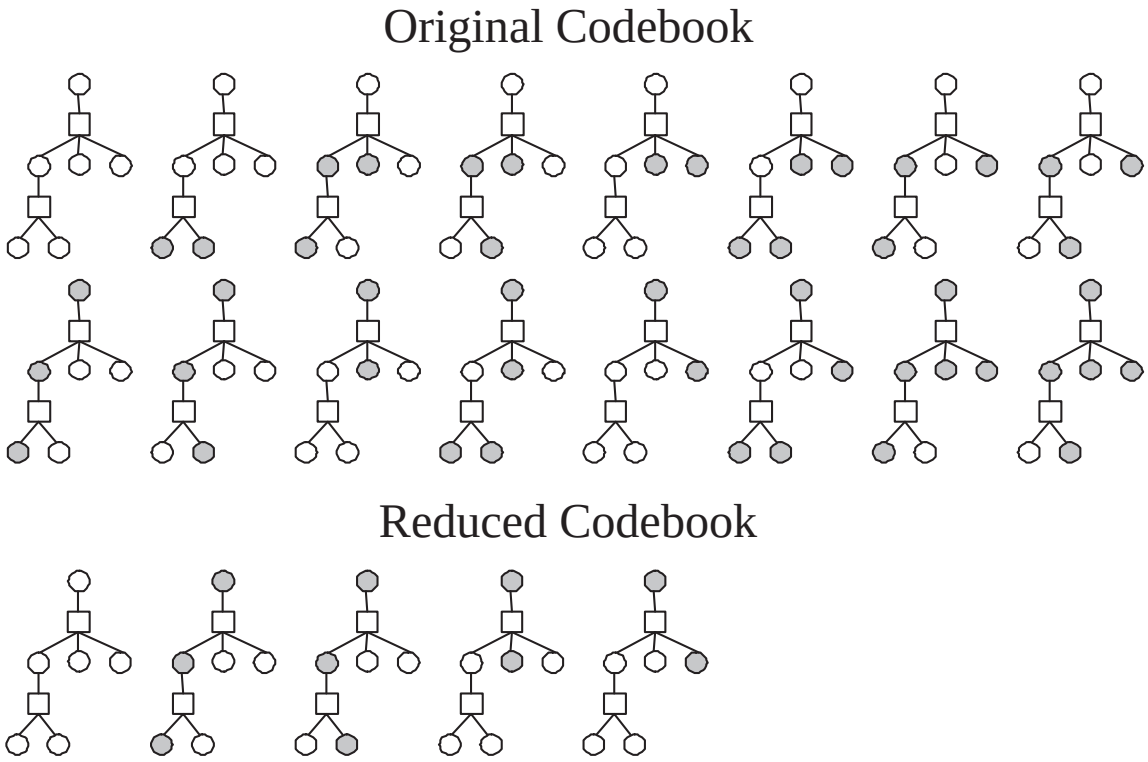


Figure 4.2: A Comparison of the original codebook $\mathcal{C}$ and the reduced codebook $\mathcal{C}_r$, where variable nodes with value 1 are denoted by filled gray circles, variable nodes with value 0 by open circles, and check nodes by squares.

2. For every check node c at this level whose parent node is labelled as "survived", it must have at least one child variable node with value 1 in $\mathbf{c}_k$ since "survived" nodes always have value 1 in $\mathbf{c}_k$. Choose an arbitrary child variable node of c with value 1 in $\mathbf{c}_k$, and label it as "survived". Then, label the *subtrees* emanating from the other unlabelled child variable nodes of c as "dropped".

3. If there are no check nodes at the next lower level of the tree, stop. Otherwise, move to the check nodes at the next lower level and go back to 2.

As we can see after this labeling procedure, the check nodes with a "survived" parent node all have exactly one "survived" child node, and the ones with a "dropped" parent node have purely "dropped" child nodes. Therefore, if we let

$$\mathbf{c}_m = \begin{cases} \mathbf{0} & \text{for all "dropped" bits,} \\ \mathbf{1} = \mathbf{c}_k & \text{for all "survived" bits,} \end{cases} \tag{4.8}$$

where $\mathbf{0}$ denotes the all-zero word and $\mathbf{1}$ denotes the all-one word, then we have $\mathbf{c}_m \in \mathcal{C}_r \subset \mathcal{C}$. Moreover, since the root bit is labelled as "survived", $c_{m1} = c_{k1} = 1$. In the following, we would like to prove that $\mathbf{y} \in \mathcal{U}_{m0}$, which completes the proof of the lemma. Let

$$\mathbf{c}_l = \begin{cases} \mathbf{c}_k & \text{for all "dropped" bits,} \\ \mathbf{0} & \text{for all "survived" bits,} \end{cases} \tag{4.9}$$

i.e., let $\mathbf{c}_l$ be the bitwise XOR of $\mathbf{c}_k$ and $\mathbf{c}_m$. Then, since $\mathbf{c}_k$ and $\mathbf{c}_m$ are valid codewords in $\mathcal{C}$, so is $\mathbf{c}_l$. Moreover, since $\mathbf{y} \in \mathcal{U}_k$ implies $\mathbf{y} \in \mathcal{U}_{kl}$, we have from (4.5)

that

$$\prod_{j=1}^{n} f(y_j|c_{kj}) \geq \prod_{j=1}^{n} f(y_j|c_{lj}) \Rightarrow \prod_{\text{all ``survived'' bits } j} f(y_j|c_{kj}) \geq \prod_{\text{all ``survived'' bits } j} f(y_j|c_{lj}) \tag{4.10a}$$

$$\Rightarrow \prod_{\text{all ``survived'' bits } j} f(y_j|c_{mj}) \geq \prod_{\text{all ``survived'' bits } j} f(y_j|0) \tag{4.10b}$$

$$\Rightarrow \prod_{j=1}^{n} f(y_j|c_{mj}) \geq \prod_{j=1}^{n} f(y_j|0), \tag{4.10c}$$

which proves that $\mathbf{y} \in \mathcal{U}_{m0}$ as desired. ■

This lemma shows that we can upper bound P_b^{MS} by the probability of MLSqD error P_s^{MLSqD} on the reduced codebook $\mathcal{C}_r$ assuming that the all-zero codeword is transmitted. One way to proceed from here is to use union bound and the fact (see [64, Theorem 7.5] for a proof) that

$$P(\mathbf{y} \in \mathcal{U}_{i0}|\mathbf{x} = \mathbf{c}_0) \leq D^{w(\mathbf{c}_i)}, \quad \forall i, \tag{4.11}$$

where $w(\mathbf{c}_i)$ denotes the Hamming weight of $\mathbf{c}_i$, and D is the Bhattacharyya parameter associated with the MBIOS channel $f(y|x)$, to further upper bound P_s^{MLSqD}. For this purpose, we would like to introduce the weight enumerator $N_l(x)$ of the nonzero codewords of the reduced codebook $\mathcal{C}_l$ of the tree code G_l of level $l+1$ associated with a randomly drawn code from the (λ, ρ) LDPC ensemble. Let A_i be the number of codewords of weight i in $\mathcal{C}_l$. We define $N_l(x)$ by

$$N_l(x) \triangleq \sum_{i=1}^{\infty} A_i x^i. \tag{4.12}$$

Moreover, let $\overline{N_l(x)}$ denote the expected value of $N_l(x)$ averaged over the whole (λ, ρ) LDPC ensemble. We have the following lemma.

Lemma 4.2 $\overline{N_0(x)} = x$ *and*

$$\overline{N_l(x)} = \lambda(\rho'(1)\overline{N_{l-1}(x)}), \quad \forall l \geq 1, \tag{4.13}$$

where $\rho'(x)$ *denotes the derivative of* $\rho(x)$ *with respect to* x.

Proof: It is obvious that $\overline{N_0(x)} = x$. To prove the recursion for $\overline{N_l(x)}$, first consider the subtree emanating from the ith check node c_i immediately below the root bit. Let $Z_l^{(i)}(x)$ denote the weight enumerator of the nonzero codewords of the reduced codebook of this subtree. Since, the root bit is 1 for all nonzero codewords in the reduced codebook $\mathcal{C}_l$, there is exactly one child subtree with root 1 emanating from c_i for all nonzero codewords in $\mathcal{C}_l$. Therefore, if c_i has degree d_c, then we have

$$Z_l^{(i)}(x) = (d_c - 1)N_{l-1}(x), \tag{4.14}$$

which implies that

$$\overline{Z_l^{(i)}(x)} = \sum_{i=1}^{\infty}(i-1)\overline{N_{l-1}(x)}\rho_i = \rho'(1)\overline{N_{l-1}(x)}. \tag{4.15}$$

Similarly, if the root bit has degree d_v, then we have

$$N_l(x) = \prod_{i=1}^{d_v-1} Z_l^{(i)}(x), \tag{4.16}$$

which implies that

$$\overline{N_l(x)} = \sum_{j=1}^{\infty} \overline{\prod_{i=1}^{d_c-1} Z_l^{(i)}(x)} \lambda_j = \sum_{j=1}^{\infty} \prod_{i=1}^{d_c-1} \overline{Z_l^{(i)}(x)} \lambda_j = \lambda(\overline{Z_l^{(1)}(x)}), \tag{4.17}$$

where the second equality follows from the fact that the subtrees emanating from different c_i's are generated independently, and the third equality follows from the fact that $\overline{Z_l^{(i)}(x)}$ does not depend on i as shown in (4.15). Combining (4.15) and (4.17), the lemma is proved. ■

Using this lemma and the union bound on P_s^{MLSqD}, we have the following theorem.

Theorem 4.1 *Given any (λ, ρ) LDPC ensemble, let P_l^{MS} and P_l^{SP} be its asymptotic (as the codeword length approaches infinity) average bit error probability after l iterations under MS and SP decoding, respectively, on an MBIOS channel with Bhattacharyya parameter D. If we define the sequence $\{z_l\}_{l=0}^{\infty}$ by $z_0 = D$, and*

$$z_l = \lambda(\rho'(1) z_{l-1}), \quad \forall l \geq 1, \tag{4.18}$$

then we have

$$P_l^{SP} \leq P_l^{MS} \leq z_l, \quad \forall l \geq 0. \tag{4.19}$$

Proof: From Lemma 4.2, we see that $z_l = \overline{N_l(D)}$ for all l. Now, the theorem follows from the union bound on P_s^{MLSqD} as discussed above. ■

A similar result to Theorem 4.1 is established in [25, Lemma 1]. However, Theorem 4.1 differs from [25, Lemma 1] in two aspects. First, Theorem 4.1 is proved for both MS and SP decoding while [25, Lemma 1] was proved only for the SP decoding. Second, we did not keep track of the evolution of the Bhattacharyya parameters, which are used in [25] to bound P_l^{SP} for all l.

4.4 Sum-Product Decoding Performance Analysis

Let

$$m = \log \frac{f(y|0)}{f(y|1)} \tag{4.20}$$

be the log-likelihood ratio (LLR) of the input variable x given the output variable y of the MBIOS channel $f(y|x)$. Moreover, let M be the random variable whose realization is m assuming $x = 0$, and $g(m)$ be the pdf of M. Then the symmetry condition (4.3) becomes

$$g(-m) = e^{-m} g(m), \quad \forall m \in \mathbb{R}. \tag{4.21}$$

Also, the Bhattacharyya parameter associated with $f(y|x)$ can be expressed as

$$B(M) = \int_{-\infty}^{\infty} f(y|0) \sqrt{\frac{f(y|1)}{f(y|0)}} dy = E\left[e^{-\frac{M}{2}}\right]. \tag{4.22}$$

Define the probability of error $P_e(M)$ of the LLR M under ML decoding as follows

$$P_e(M) = \int_{-\infty}^{0} g(m) dm. \tag{4.23}$$

We have from (4.21) that

$$B(M) = \int_{-\infty}^{\infty} g(m) e^{-\frac{m}{2}} dm = 2 \int_{-\infty}^{0} g(m) e^{-\frac{m}{2}} dm \geq 2 P_e(M). \tag{4.24}$$

Also, we have the following lemma.

Lemma 4.3

$$E\left[\tanh\frac{|M|}{2}\right] = 1 - 2P_e(M) \tag{4.25}$$

Proof:

$$\begin{aligned} E\left[\tanh\frac{|M|}{2}\right] &= \int_0^\infty \left(\tanh\frac{m}{2}\right)[g(m) + g(-m)]\,dm \\ &= \int_0^\infty \left(\frac{1-e^{-m}}{1+e^{-m}}\right)\left[g(m) + e^{-m}g(m)\right]dm \\ &= \int_0^\infty \left(1-e^{-m}\right)g(m)dm \\ &= \int_0^\infty g(m)dm - \int_{-\infty}^0 g(m)dm \\ &= 1 - 2P_e(M) \end{aligned} \tag{4.26}$$

■

Now, consider the SP decoding on a tree code used on an MBIOS channel. As shown in [15], all the SP decoding messages can be represented by the LLR's and satisfy the symmetry condition (4.21). Assuming that the all-zero codeword is transmitted, we have the following lemma characterizing the evolution of the Bhattacharyya parameters and the probability of ML decoding errors associated with the incoming and outgoing messages at a variable node. Note that in the following, we will use capital letters to denote random variables, whose realizations are denoted by the corresponding lower-case letters.

Lemma 4.4 *Let v be a variable node of degree d_v. Furthermore, let M_0 be the incoming message from the channel, M_v the outgoing message on an edge, and $M_1, M_2, \ldots, M_{d_v-1}$ the incoming messages from the other edges. Assuming all the*

incoming messages are independent with each other, we have

$$B(M_v) = \prod_{i=0}^{d_v-1} B(M_i), \tag{4.27}$$

and

$$2P_e(M_v) \geq \prod_{i=0}^{d_v-1} \left[2P_e(M_i)\right]. \tag{4.28}$$

Proof: Since under SP decoding [15],

$$m_v = \sum_{i=0}^{d_v-1} m_i, \tag{4.29}$$

it suffices to prove the lemma for $d_v = 2$. The general statements then follow by induction. From (4.22), (4.27) is true since the incoming messages are independent with each other. Now, we prove (4.28) for $d_v = 2$.

From Lemma 4.3, we have

$$\begin{aligned}
&2P_e(M_v) \geq \prod_{i=0}^{1} \left[2P_e(M_i)\right] \\
\Leftrightarrow &1 - E\left[\tanh\frac{|M_v|}{2}\right] \geq \left(1 - E\left[\tanh\frac{|M_0|}{2}\right]\right)\left(1 - E\left[\tanh\frac{|M_1|}{2}\right]\right) \qquad (4.30a) \\
\Leftrightarrow &E\left[\tanh\frac{|M_v|}{2}\right] \leq \\
&E\left[\tanh\frac{|M_0|}{2}\right] + E\left[\tanh\frac{|M_1|}{2}\right] - E\left[\tanh\frac{|M_0|}{2}\right] E\left[\tanh\frac{|M_1|}{2}\right]. \qquad (4.30b)
\end{aligned}$$

Since $m_v = m_0 + m_1$, the left hand side of (4.30b) becomes

$$\begin{aligned}
&\int_{m_0+m_1\geq 0} \left[1 - e^{-(m_0+m_1)}\right] g_0(m_0)g_1(m_1)dm_0dm_1 \\
=&\int_{m_0\geq 0, m_1\geq 0} \left[1 - e^{-(m_0+m_1)}\right] g_0(m_0)g_1(m_1)dm_0dm_1 \\
&+ \int_{m_0\geq -m_1, m_1<0} \left[1 - e^{-(m_0+m_1)}\right] g_0(m_0)g_1(m_1)dm_0dm_1 \\
&+ \int_{m_1\geq -m_0, m_0<0} \left[1 - e^{-(m_0+m_1)}\right] g_0(m_0)g_1(m_1)dm_0dm_1, \qquad (4.31)
\end{aligned}$$

where g_0 and g_1 are the pdf's of M_0 and M_1, respectively. On the other hand, the right hand side of (4.30b) becomes

$$\begin{aligned}
&\int_{m_0\geq 0} \left(1 - e^{-m_0}\right) g_0(m_0)dm_0 + \int_{m_1\geq 0} \left(1 - e^{-m_1}\right) g_1(m_1)dm_1 \\
&- \int_{m_0\geq 0, m_1\geq 0} \left(1 - e^{-m_0}\right)\left(1 - e^{-m_1}\right) g_0(m_0)g_1(m_1)dm_0dm_1 \\
=&\int_{m_0\geq 0, m_1\geq 0} \left[1 - e^{-(m_0+m_1)}\right] g_0(m_0)g_1(m_1)dm_0dm_1 \\
&+ \int_{m_0\geq 0, m_1<0} \left(1 - e^{-m_0}\right) g_0(m_0)g_1(m_1)dm_0dm_1 \\
&+ \int_{m_1\geq 0, m_0<0} \left(1 - e^{-m_1}\right) g_0(m_0)g_1(m_1)dm_0dm_1. \qquad (4.32)
\end{aligned}$$

Comparing the right hand sides of (4.31) and (4.32), since

$$\begin{aligned}
&\int_{m_i\geq 0, m_j<0} \left(1 - e^{-m_i}\right) g_i(m_i)g_j(m_j)dm_idm_j \\
\geq&\int_{m_i\geq -m_j, m_j<0} \left(1 - e^{-m_i}\right) g_i(m_i)g_j(m_j)dm_idm_j \qquad (4.33a) \\
\geq&\int_{m_i\geq -m_j, m_j<0} \left(1 - e^{-m_i}e^{-m_j}\right) g_i(m_i)g_j(m_j)dm_idm_j, \qquad (4.33b)
\end{aligned}$$

for (i, j) equal $(1, 2)$ and $(2, 1)$, (4.30b) is true, and hence (4.28) is proved. ■

Similar evolution equation and inequality for the Bhattacharyya parameters and the probability of ML decoding errors at a check node are also derived in the following lemma.

Lemma 4.5 *Let c be a check node of degree d_c. Furthermore, let M_c be the outgoing message on an edge, and $M_1, M_2, \ldots, M_{d_c-1}$ the incoming messages from the other edges. Assuming all the incoming messages are independent with each other, we have*

$$1 - B(M_c) \geq \prod_{i=1}^{d_c-1} \left[1 - B(M_i)\right], \tag{4.34}$$

and

$$1 - 2P_e(M_c) = \prod_{i=1}^{d_c-1} \left[1 - 2P_e(M_i)\right]. \tag{4.35}$$

Proof: Since under SP decoding [15],

$$\tanh \frac{m_c}{2} = \prod_{i=1}^{d_c-1} \tanh \frac{m_i}{2}, \tag{4.36}$$

and from (4.2), $1-B(M) \geq 0$ for all LLR M, it again suffices to prove the lemma for $d_c = 3$. The general statements then follow by induction. From Lemma 4.3, (4.36), and the fact that

$$|\tanh x| = \tanh |x|, \quad \forall x, \tag{4.37}$$

(4.35) follows directly by the independence of the incoming messages. Now, we prove (4.34) for $d_c = 3$, which, from (4.22), is equivalent to proving that

$$E\left[e^{-\frac{M_c}{2}}\right] \leq 1 - \prod_{i=1}^{2} \left(1 - E\left[e^{-\frac{M_i}{2}}\right]\right). \tag{4.38}$$

From (4.36), we have

$$m_c = \log\left(\frac{1+\prod_{i=1}^{2}\tanh\frac{m_i}{2}}{1-\prod_{i=1}^{2}\tanh\frac{m_i}{2}}\right) \tag{4.39a}$$

$$= \log\left(\frac{1+\prod_{i=1}^{2}\frac{1-e^{-m_i}}{1+e^{-m_i}}}{1-\prod_{i=1}^{2}\frac{1-e^{-m_i}}{1+e^{-m_i}}}\right) \tag{4.39b}$$

$$= \log\left(\frac{\prod_{i=1}^{2}(1+e^{-m_i})+\prod_{i=1}^{2}(1-e^{-m_i})}{\prod_{i=1}^{2}(1+e^{-m_i})-\prod_{i=1}^{2}(1-e^{-m_i})}\right) \tag{4.39c}$$

$$= \log\left(\frac{1+e^{-m_1}e^{-m_2}}{e^{-m_1}+e^{-m_2}}\right). \tag{4.39d}$$

Let g_1 and g_2 be the pdf's of M_1 and M_2, respectively. From (4.39), we have on the left hand side of (4.38) that

$$E\left[e^{-\frac{M_c}{2}}\right] = E\left[\sqrt{\frac{e^{-M_1}+e^{-M_2}}{1+e^{-M_1}e^{-M_2}}}\right] \tag{4.40a}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_1(m_1)g_2(m_2)\sqrt{\frac{e^{-m_1}+e^{-m_2}}{1+e^{-m_1}e^{-m_2}}}dm_1dm_2 \tag{4.40b}$$

$$= \int_{0}^{\infty}\int_{0}^{\infty} [g_1(m_1)g_2(m_2)+g_1(-m_1)g_2(-m_2)]\sqrt{\frac{e^{-m_1}+e^{-m_2}}{1+e^{-m_1}e^{-m_2}}}$$

$$+ [g_1(m_1)g_2(-m_2)+g_1(-m_1)g_2(m_2)]\sqrt{\frac{1+e^{-m_1}e^{-m_2}}{e^{-m_1}+e^{-m_2}}}dm_1dm_2 \tag{4.40c}$$

$$= 2\int_{0}^{\infty}\int_{0}^{\infty} g_1(m_1)g_2(m_2)\sqrt{(e^{-m_1}+e^{-m_2})(1+e^{-m_1}e^{-m_2})}dm_1dm_2, \tag{4.40d}$$

where the last equality follows from (4.21). Similarly, the right hand side of (4.38) can be written as follows:

$$
\begin{aligned}
&1-\prod_{i=1}^{2}\left(1-E\left[e^{-\frac{M_i}{2}}\right]\right)\\
=&E\left[1-\left(1-e^{-\frac{M_1}{2}}\right)\left(1-e^{-\frac{M_2}{2}}\right)\right] &\text{(4.41a)}\\
=&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_1(m_1)g_2(m_2)\left(e^{-\frac{m_1}{2}}+e^{-\frac{m_2}{2}}-e^{-\frac{(m_1+m_2)}{2}}\right)dm_1dm_2 &\text{(4.41b)}\\
=&\int_{0}^{\infty}\int_{0}^{\infty}[g_1(m_1)e^{-\frac{m_1}{2}}+g_1(-m_1)e^{\frac{m_1}{2}}][g_2(m_2)+g_2(-m_2)]\\
&+[g_1(m_1)+g_1(-m_1)][g_2(m_2)e^{-\frac{m_2}{2}}+g_2(-m_2)e^{\frac{m_2}{2}}]\\
&-[g_1(m_1)e^{-\frac{m_1}{2}}+g_1(-m_1)e^{\frac{m_1}{2}}][g_2(m_2)e^{-\frac{m_2}{2}}+g_2(-m_2)e^{\frac{m_2}{2}}]dm_1dm_2 &\text{(4.41c)}\\
=&2\int_{0}^{\infty}\int_{0}^{\infty} g_1(m_1)g_2(m_2)\\
&\left[e^{-\frac{m_1}{2}}\left(1+e^{-m_2}\right)+e^{-\frac{m_2}{2}}\left(1+e^{-m_1}\right)-2e^{-\frac{m_1}{2}}e^{-\frac{m_2}{2}}\right]dm_1dm_2, &\text{(4.41d)}
\end{aligned}
$$

where the first equality follows from the independence of M_1 and M_2, and the last equality follows from (4.21). Let $a_i \triangleq e^{-m_i}$, for $i = 1, 2$. Then, we see from (4.40) and (4.41) that it is sufficient to prove

$$\sqrt{(1+a_1a_2)(a_1+a_2)} \leq (1+a_2)\sqrt{a_1}+(1+a_1)\sqrt{a_2}-2\sqrt{a_1a_2}, \tag{4.42}$$

for all $0 \leq a_i \leq 1$, for all i. Now, since

$$\begin{aligned}
&\left[(1+a_2)\sqrt{a_1}+(1+a_1)\sqrt{a_2}-2\sqrt{a_1a_2}\right]^2-(1+a_1a_2)(a_1+a_2)\\
=&(1+a_2)^2a_1+(1+a_1)^2a_2+4a_1a_2+2(1+a_1)(1+a_2)\sqrt{a_1a_2}\\
&-4(1+a_1)a_2\sqrt{a_1}-4(1+a_2)a_1\sqrt{a_2}-(1+a_2^2)a_1-(1+a_1^2)a_2 &\text{(4.43a)}\\
=&2\sqrt{a_1a_2}\left[4\sqrt{a_1a_2}+(1+a_1)(1+a_2)-2(1+a_1)\sqrt{a_2}-2(1+a_2)\sqrt{a_1}\right] &\text{(4.43b)}\\
=&2\sqrt{a_1a_2}\left[(1+a_1)-2\sqrt{a_1}\right]\left[(1+a_2)-2\sqrt{a_2}\right] &\text{(4.43c)}\\
=&2\sqrt{a_1a_2}(1-\sqrt{a_1})^2(1-\sqrt{a_2})^2 &\text{(4.43d)}\\
\geq&0, &\text{(4.43e)}
\end{aligned}$$

(4.42) is true, and hence (4.34) is proved. ■

Since all the incoming messages to the variable and check nodes are independent with each other on a tree code under SP decoding, Lemma 4.4 and Lemma 4.5 can be used to imply the following theorem.

Theorem 4.2 *For the (λ, ρ) LDPC ensemble with SP decoding used on an MBIOS channel with conditional pdf $f(y|x)$, the Bhattacharyya parameter B_l associated with the outgoing message of any variable node after the lth decoding iteration asymptotically satisfies*

$$B_l \leq D\lambda\left(1-\rho\left(1-B_{l-1}\right)\right), \tag{4.44}$$

where

$$D \triangleq \int_{-\infty}^{\infty}\sqrt{f(y|0)f(y|1)}dy, \tag{4.45}$$

and $B_0 = D$. Moreover, the probability of bit error P_l of any variable node after the

lth decoding iteration asymptotically satisfies

$$2P_l \geq 2P_0\lambda\left(1-\rho\left(1-2P_{l-1}\right)\right), \tag{4.46}$$

where P_0 is the uncoded bit error probability under ML decoding of the channel.

Proof: As discussed in Section 4.2, the probability of error associated with the outgoing message of any variable node after the lth decoding iteration is, asymptotically as the codeword length goes to infinity, the one of the root variable node of a tree of level $l+1$. Since all the variable and check nodes in the tree have exactly the same degree distributions λ and ρ, respectively, and the channel is memoryless, all the incoming messages from child nodes to a parent node are independent and identically distributed. Hence, if we let v be a variable node in the tree, M_v its outgoing message, M_0 the incoming message from the channel, and M_c one of its incoming messages from its child nodes, then we have from Lemma 4.4 that

$$B(M_v) = B(M_0)\sum_{i=1}^{\infty}\left[B(M_c)\right]^{(i-1)}\lambda_i = D\lambda(B(M_c)), \tag{4.47}$$

and

$$2P_e(M_v) \geq 2P_e(M_0)\sum_{i=1}^{\infty}\left[2P_e(M_c)\right]^{(i-1)}\lambda_i = 2P_e(M_0)\lambda(2P_e(M_c)). \tag{4.48}$$

Similarly, if we let c be a check node in the tree, M_c its outgoing message, and M_v one of its incoming messages, then we have from Lemma 4.5 that

$$B(M_c) \leq 1-\sum_{i=1}^{\infty}\left[1-B(M_v)\right]^{(i-1)}\rho_i = 1-\rho(1-B(M_v)), \tag{4.49}$$

and

$$2P_e(M_c) = 1 - \sum_{i=1}^{\infty} \left[1 - 2P_e(M_v)\right]^{(i-1)} \rho_i = 1 - \rho(1 - 2P_e(M_v)). \tag{4.50}$$

Combining (4.47), (4.48), (4.49), (4.50), and the fact that λ and ρ are monotonically increasing functions (since all λ_i's and ρ_i's are nonnegative), the theorem is proved. ■

The recursion (4.44) was also found independently in [30, Theorem 2]. Notice that on the BEC of erasure probability ϵ, the Bhattacharyya parameter of this channel is ϵ, and the probability of uncoded bit error is $\epsilon/2$. Hence, (4.44) and (4.46) are both satisfied with equality, and recover the well-known DE equation

$$x_l = \epsilon\lambda\left(1 - \rho\left(1 - x_{l-1}\right)\right), \quad x_0 = \epsilon, \tag{4.51}$$

where x_l is the bit erasure probability after the lth decoding iteration on the BEC. Using this fact, we have the following corollary.

Corollary 4.1 *For any (λ, ρ) LDPC ensemble, its asymptotic SP decoding performance on the MBIOS channel with Bhattacharyya parameter D is always better than or equal to that on the BEC with erasure probability $\epsilon \geq D$. Moreover, its asymptotic SP decoding performance on the MBIOS channel with uncoded bit error probability P_0 is always worse than or equal to that on the BEC with erasure probability $\epsilon \leq 2P_0$.*

Proof: From (4.44), (4.46) and (4.51), we have $x_l \geq B_l$ when $\epsilon \geq D$, and $x_l \leq 2P_l$ when $\epsilon \leq 2P_0$ for all $l \geq 0$. Since the erasure probability is two times the bit error probability, which in general is less than or equal to the Bhattacharyya parameter as shown in (4.24), the corollary is proved. ■

Notice that, Lemma 4.4 and Lemma 4.5 can also be used for the more general family of multi-edge type LDPC codes [16], including the IRA codes [17,18] and the LDPC-GM codes introduced in Chapter 3, to produce similar results. Hence, Corollary 4.1 is not restricted to the irregular LDPC codes, but also holds for the general multi-edge type LDPC codes.

4.5 Conclusion

In this chapter, we analyze the asymptotic performance of LDPC codes under MS and SP decoding on MBIOS channels. This is done by bounding the bit error probability of the root bit of the tree code associated with the (λ, ρ) LDPC ensemble assuming that the all-zero codeword is transmitted. When MS decoding is performed on this tree code, we upper bound the probability of the root bit being in error by the probability of sequence error under MLSqD of a subcode of the tree code. A recursive equation describing the evolution of the weight enumerator of this subcode after each iteration is then derived and used in a union bound to bound the ML decoded sequence error of this subcode. As a result, we obtain a recursive upper bound on the bit error probability after each iteration for the LDPC codes under MS decoding on MBIOS channels. Note that this upper bound is also an upper bound for the SP decoding since SP decoding is optimal on the bit error probability for tree codes. This result is very similar to [25, Lemma 1] with the difference being that we establish it not only for the SP decoding, but also for the MS decoding, and that we obtain it via a totally different approach.

When SP decoding performance is considered, we derive a recursive upper bound on the Bhattacharyya parameter as well as a recursive lower bound on the probability of bit error associated with the outgoing message of the root bit after each iteration. More significantly, both recursions recover the DE equation on the BEC for LDPC

codes with the inequalities being exact equalities. This further implies that the SP decoding performance of LDPC codes on the BEC can serve as a lower bound of the ones on all MBIOS channels with the same Bhattacharyya parameter, and also an upper bound of the ones on all MBIOS channels with the same probability of uncoded bit error. This result is also true for the more general multi-edge type LDPC codes, including IRA and LDPC-GM codes, since the main ingredient in the proof, i.e., Lemma 4.4 and Lemma 4.5, can also be utilized for these codes. Note also that the derived lower bound on the probability of bit error is also valid under MS decoding due to the optimality of SP decoding on the bit error probability for tree codes.

CHAPTER 5

Low Complexity Algorithms for Joint Data Detection and Frequency/Phase Estimation

5.1 Introduction

Coherent data communication requires perfect knowledge of the frequency and phase of the carrier signal. In practical communication systems, however, the mobility of the transmitter/receiver, in conjunction with the ambient and electronic noise at the receiver circuitry results in a time-varying phase that is unknown to the receiver. The presence of this frequency/phase jitter has multiple effects on the transmitted signal, the most severe of which is distortion of the transmitted signal and generation of inter-symbol interference (ISI), which becomes significant as the frequency/phase dynamics increase. Nevertheless, even when the channel dynamics are slow, the effect is a multiplicative phase distortion that can cause a significant performance loss, if not accounted for at the receiver.

There are three basic techniques (see [65] for a tutorial review) for detecting data in the presence of frequency/phase jitter:

(i) A training sequence is transmitted periodically that aids at frequency and phase

estimation, followed by coherent data detection.

(ii) The received signal is passed through a memoryless nonlinearity (e.g., squaring binary phase shift keying (BPSK) signals,) that eliminates the data dependence, thus allowing channel estimation, followed by coherent data detection

(iii) Joint data detection and frequency/phase estimation.

Clearly, the first two schemes do not exploit all the channel information available at the received sequence. As a result, the first two solutions are adequate for high signal-to-noise ratios (SNRs) and slow channel dynamics, but they perform very poorly (in terms of bandwidth efficiency or bit-error-rate (BER)) in more severe scenarios. Thus, when high-performance codes are employed over fast channels, some sort of joint detection and estimation needs to be performed.

Motivated by the desire to exploit the information from the whole sequence while keeping a low computational complexity, several joint detection/estimation algorithms have been proposed in the literature. These can be classified into the following two categories. The first category involves alternate maximization (or more formally, expectation maximization (EM)), where the tasks of channel-conditioned data detection and data-conditioned channel estimation are performed iteratively starting from an initial channel estimate, until convergence occurs [66–68]. The second category of algorithms is based on a suboptimal search over the set of all sequences by appropriately pruning the sequence tree according to a tree-pruning algorithm, such as the T-algorithm [69], the M-algorithm [70], or the per-survivor processing (PSP) algorithm [71–75]. It is noted that the above mentioned approaches are valid for any joint data detection and channel estimation problem and not only for the particular problem of joint data detection and frequency/phase estimation discussed in this chapter. The impetus for the aforementioned research on approximate algorithms

was the belief that the *exact* solution of the joint data detection and parameter estimation problem requires an exponential complexity with respect to the sequence length to be found.

Since future communication systems will operate close to their theoretical limits, joint data detection and parameter estimation will become indispensable in achieving the highest possible performance with the given resources, and thus, the following questions arise:

(i) How accurate is the conventional wisdom that *exact* joint data detection and frequency/phase estimation requires exponential complexity with respect to the sequence length?

(ii) What is the impact of a negative answer to the above question on the design of approximate algorithms?

(iii) How can the complexity/performance tradeoff of these approximate algorithms be analyzed, and how can it be improved?

In this chapter, we continue the work initiated in [31] in trying to provide answers to the above questions. In particular, it is shown here that the exact, generalized-likelihood-based, joint detection and frequency/phase estimation of uncoded sequences is a polynomial-complexity problem in the sequence length. Furthermore, the proposed technique can be generalized to solve the problem of symbol-by-symbol soft-decision generation implied by the min-sum algorithm, which can be used when turbo-like coded sequences are utilized. In this chapter, we concentrate on the uncoded sequence detection problem, since the extension to symbol-by-symbol soft-decision generation can be performed in a way similar to the one described in [31]. Furthermore, based on the proposed exact solution, we develop a class of approximate algorithms. Finally, a framework for analyzing the performance of both exact

and approximate algorithms at arbitrary SNR is proposed. In the case of performance analysis for the exact algorithms, it is shown that analysis is possible exactly due to the novel polynomial-complexity structure. For the approximate algorithms, it is shown that their performance can get close to that of the exact algorithms with reasonable complexity.

The remaining of the chapter is structured as follows. Section 5.2 develops the system and channel model under consideration. The polynomial-complexity exact algorithm for joint data detection and frequency/phase estimation is developed in Section 5.3, after reviewing the corresponding results from [31]. Section 5.4 presents the performance analysis for the exact, as well as several approximate algorithms. Numerical results are presented in Section 5.5, while the conclusions are summarized in Section 5.6. For the sake of clarity, most of the proofs are relegated to Appendix C.

5.2 Channel Model

Consider the transmission of a length-N sequence of symbols $s_k \in \mathcal{A}$, where $\mathcal{A}$ is a set of complex-valued numbers with unit magnitude. The equivalent low-pass transmitted signal is of the form

$$s(t) = \sum_{k=1}^{N} s_k \sqrt{E_k} q(t - kT), \tag{5.1}$$

where $q(t)$ is a pulse shape function satisfying the no-ISI Nyquist criterion [76, Section 9.2.1] with unit energy, T is the symbol duration, and the energy of the kth transmitted symbol E_k equals to E_s or E_p depending on whether s_k is an information symbol or pilot symbol, respectively. If this signal is transmitted over an additive white Gaussian noise (AWGN) channel and is further rotated by a phase process $\phi(t)$ unknown to both transmitter and receiver, then the received signal $z(t)$

can be modelled as

$$z(t) = s(t)e^{j\phi(t)} + n(t), \tag{5.2}$$

where $n(t)$ is a zero mean complex white Gaussian process with one-sided power spectral density level N_0. This channel is motivated by considering the front-end of a wireless communication system, where $\phi(t)$ is due to the phase difference between the transmitted signal carrier and the mixing oscillator at the receiver. Moreover, a generally time-varying $\phi(t)$ may also be attributed to the instability of the oscillators and the mobility at both the transmitting and receiving end. However, in this model, amplitude variations are ignored. This is done mainly in order to isolate the effects of phase rotation on the system performance and is under the assumption that an automatic gain control mechanism is present that compensates for the amplitude variations.

For this observation model, optimal pre-processing depends on the phase process $\phi(t)$ and in general should employ fractionally-spaced sampling. However, for simplicity, we will assume symbol-spaced sampling, since all algorithms presented in this chapter generalize easily to the fractionally-spaced model. We will further assume perfect epoch synchronization. Then, if the phase rotation process $\phi(t)$ is modelled by a constant unknown phase $\theta \in [0, 2\pi)$, i.e., if

$$\phi(t) = \theta \quad \forall t \in [0, NT), \tag{5.3}$$

then there will be no ISI and we have the following equivalent discrete-time model

$$\mathbf{z} = \mathbf{D}\mathbf{s}e^{j\theta} + \mathbf{n}, \tag{5.4}$$

where $\mathbf{s} \triangleq (s_1, s_2, \ldots, s_N)^T$ as the superscript T denotes transpose, $\mathbf{D}$ is a diagonal matrix with diagonal elements $(\sqrt{E_1}, \sqrt{E_2}, \ldots, \sqrt{E_N})$, $\mathbf{z} \triangleq (z_1, z_2, \ldots, z_N)^T$ is the vector of symbol-spaced observations, and $\mathbf{n} \triangleq (n_1, n_2, \ldots, n_N)^T$ is a vector of independent identically-distributed (iid) zero-mean circularly symmetric complex Gaussian random variables with variance $N_0/2$ per real and imaginary component. On the other hand, if the phase rotation process is modelled by

$$\phi(t) = 2\pi(f_d/T)t + \theta \quad \forall t \in [0, NT), \tag{5.5}$$

where $f_d \in [0, 1)$ is the normalized frequency jitter, and $\theta \in [0, 2\pi)$ is the phase shift, then as shown in [65] under the assumption of no ISI and perfect gain control, the discrete-time model becomes

$$z_k = s_k\sqrt{E_k}e^{j(2\pi f_d k+\theta)} + n_k \quad \forall k = 1, 2, \ldots, N. \tag{5.6}$$

5.3 Algorithms for Exact Generalized-likelihood Detection

For this section and the rest of the chapter, we will restrict our attention to the special case where we use equally probable antipodal signaling, i.e., $\mathcal{A} = \{+1, -1\}$ for simplicity. Moreover, we will assume that the first one, and the first two symbols are pilot symbols for models (5.4), and (5.6), respectively. With a little abuse of notation, it is understood that all following detection algorithms are applied only to information symbols by setting beforehand the pilot symbols to +1. However, we note that all the algorithms and results presented in this chapter can be generalized to arbitrary alphabet $\mathcal{A}$, unequal a-priori probabilities, arbitrary pilot symbol positioning, and energy assignments with a complexity increase by a factor $O(|\mathcal{A}|^2)$.

5.3.1 Background: The Constant Phase Model

In this subsection, we review the low complexity exact generalized-likelihood algorithm proposed in [31, 77, 78] for completeness and for the purpose of showing the basic idea from which we develop the exact algorithm for the more complicated model (5.6). The sequence estimate based on the generalized-likelihood ratio test (GLRT) for model (5.4) can be written in the following double maximization form

$$\hat{\mathbf{s}}_{GLRT} \triangleq \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} \left\{ \max_{\tilde{\theta}\in[0,2\pi)} p(\mathbf{z}|\tilde{\mathbf{s}},\tilde{\theta}) \right\} = \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} \left\{ \max_{\tilde{\theta}\in[0,2\pi)} \Re\{\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}e^{j\tilde{\theta}}\} \right\} \tag{5.7a}$$

$$= \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} |\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}|. \tag{5.7b}$$

where $\Re\{z\}$ denotes the real part of the complex number z and $\mathbf{z}^H$ the Hermitian transpose of the vector $\mathbf{z}$. If we assume that the constant phase rotation θ is uniformly distributed in $[0, 2\pi)$, then the GLRT solution coincides with the maximum-likelihood sequence detection (MLSD) solution $\hat{\mathbf{s}}_{MLSD}$ [31]. Furthermore, this optimal solution can be obtained exactly with $O(N \log N)$ complexity as follows (for details on the correctness of the algorithm, refer to [31]).

1. Calculate the set of partitioning points (or thresholds) from the received signal $\mathbf{z}$ as follows

$$\Phi \triangleq \{\phi_i \in [0, 2\pi) : \phi_i \triangleq \angle z_i \pm \frac{\pi}{2}, \quad \forall i = 2, 3, \ldots, N\}, \tag{5.8}$$

   where $\angle z$ denotes the phase component of the complex number z.

2. Sort Φ so that $\Phi = \{0 \le t_1 \le t_2 \le \ldots \le t_{2(N-1)} \le 2\pi\}$. We use $j(i)$ to denote the index of the observation symbol $z_{j(i)}$ associated with t_i.

3. Find the candidate sequence supported by the first partition, which contains

phase 0,

$$\mathbf{s}_1 = \hat{\mathbf{s}}(0), \tag{5.9}$$

where

$$\hat{\mathbf{s}}(\tilde{\theta}) \triangleq \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} \Re\{\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}e^{j\tilde{\theta}}\}. \tag{5.10}$$

4. Sequentially evaluate the candidate sequence $\mathbf{s}_{i+1}$ from $\mathbf{s}_i$ by flipping the $j(i)$th bit of $\mathbf{s}_i$.

5. As explicitly proved in [31], $\hat{\mathbf{s}}_{GLRT}$ must be one of the candidate sequences $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{2(N-1)}$. Hence we have

$$\hat{\mathbf{s}}_{GLRT} = \arg\max_{\tilde{\mathbf{s}}\in\{\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_{2(N-1)}\}} \Re\{\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}e^{j\hat{\theta}(\tilde{\mathbf{s}})}\} = \arg\max_{\tilde{\mathbf{s}}\in\{\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_{2(N-1)}\}} |\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}|, \tag{5.11}$$

where

$$\hat{\theta}(\tilde{\mathbf{s}}) \triangleq \arg\max_{\tilde{\theta}\in[0,2\pi)} \Re\{\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}e^{j\tilde{\theta}}\} = -\angle(\mathbf{z}^H\mathbf{D}\tilde{\mathbf{s}}). \tag{5.12}$$

The actual algorithm in [31, 77, 78] further simplifies the metric evaluation in (5.11), so that the overall complexity is dominated by the sorting in step 2, which can be performed with $O(N \log N)$ complexity. Note that the basic idea behind the algorithm is to partition the parameter space $[0, 2\pi)$ in such a way that performing coherent sequence detection with one hypothesized parameter taken from each set of the partition provides a sufficient set of candidate sequences for finding the GLRT solution.

5.3.2 The Linear Phase Model

Consider the model in (5.6). If the statistical models of the unknown parameters f_d and θ are known to the receiver, then we have the following optimal MLSD solution

$$\hat{\mathbf{s}}_{MLSD} \triangleq \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} p(\mathbf{z}|\tilde{\mathbf{s}}) = \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} E_{f_d,\theta}\left[p(\mathbf{z}|\tilde{\mathbf{s}}, f_d, \theta)\right]. \tag{5.13}$$

However, if such a statistical model does not exist or is unknown to the receiver, and thereby the optimality of detection rules is not defined, we can apply the following sensible GLRT criterion for detection

$$\hat{\mathbf{s}}_{GLRT} \triangleq \arg\max_{\tilde{\mathbf{s}}\in\mathcal{A}^N} \left\{ \max_{(\tilde{f}_d,\tilde{\theta})\in\Lambda} p(\mathbf{z}|\tilde{\mathbf{s}}, \tilde{f}_d, \tilde{\theta}) \right\}, \tag{5.14}$$

where $\Lambda \triangleq [0,1)\times[0,2\pi)$ is the 2-dimensional parameter space over which maximization of the unknown parameters f_d and θ is performed for each possible transmitted sequence. In the rest of the section, we will aim at finding this exact GLRT solution with polynomial complexity.

Defining

$$\hat{f}_d(\tilde{\mathbf{s}}) \triangleq \arg\max_{\tilde{f}_d\in[0,1)} \left\{ \max_{\tilde{\theta}\in[0,2\pi)} p(\mathbf{z}|\tilde{\mathbf{s}}, \tilde{f}_d, \tilde{\theta}) \right\} = \arg\max_{\tilde{f}_d\in[0,1)} \left| \sum_{k=1}^{N} z_k^* \tilde{s}_k \sqrt{E_k} e^{j2\pi\tilde{f}_d k} \right| \tag{5.15a}$$

and

$$\hat{\theta}(\tilde{\mathbf{s}}) \triangleq \arg\max_{\tilde{\theta}\in[0,2\pi)} \left\{ \max_{\tilde{f}_d\in[0,1)} p(\mathbf{z}|\tilde{\mathbf{s}}, \tilde{f}_d, \tilde{\theta}) \right\} = -\angle\left(\sum_{k=1}^{N} z_k^* \tilde{s}_k \sqrt{E_k} e^{j2\pi\hat{f}_d(\tilde{\mathbf{s}})k} \right) \tag{5.15b}$$

to be the sequence-conditioned frequency and phase estimates, respectively, where the superscript $*$ denotes complexity conjugate, and $\tilde{s}_k$ denotes the kth component

of the vector $\tilde{\mathbf{s}}$, we have

$$\hat{\mathbf{s}}_{GLRT} = \arg \max_{\tilde{\mathbf{s}} \in \mathcal{A}^N} p(\mathbf{z}|\tilde{\mathbf{s}}, \hat{f}_d(\tilde{\mathbf{s}}), \hat{\theta}(\tilde{\mathbf{s}})). \tag{5.16}$$

Looking at (5.16), we can make the following observation. There are two sources of complexity in finding the GLRT solution. The first one is combinatorial in nature and is related to the exponential growth of the size of $\mathcal{A}^N$, over which a maximization is taken. The second source of complexity is computational in nature and is related to the evaluation of each metric $p(\mathbf{z}|\tilde{\mathbf{s}}, \hat{f}_d(\tilde{\mathbf{s}}), \hat{\theta}(\tilde{\mathbf{s}}))$ for a given hypothesized sequence. This latter complexity is linear in N for the constant phase model as evidenced in (5.7b). It should be clear however, that there is no hope in finding a closed-form solution for the metric $p(\mathbf{z}|\tilde{\mathbf{s}}, \hat{f}_d(\tilde{\mathbf{s}}), \hat{\theta}(\tilde{\mathbf{s}}))$, since this is equivalent to finding a closed-form solution to the sequence-conditioned frequency estimate $\hat{f}_d(\tilde{\mathbf{s}})$ in (5.15a). But this, in turn, involves maximization of the magnitude of the discrete-time Fourier transform (DTFT) of the sequence $\{z_k \tilde{s}_k^* \sqrt{E_k}\}_{k=1}^N$ over the continuous interval $[0, 1)$ [65]. Therefore, frequency estimation can only be performed within a pre-specified accuracy with finite complexity in all practical algorithms. Since this complexity is unavoidable even when the data sequence is known, we are only interested in the first source of complexity manifesting itself through the combinatorial explosion of the number of data sequences. As can be seen in (5.16), even if we omit the frequency-estimation complexity, the exact GLRT solution still appears to demand $O(2^N)$ complexity. We will now propose an exact algorithm, which performs this task with $O(N^4)$ complexity regardless of the SNR.

Defining the parameter-conditioned sequence estimate as

$$\hat{\mathbf{s}}(\tilde{f}_d, \tilde{\theta}) \triangleq \arg \max_{\tilde{\mathbf{s}} \in \mathcal{A}^N} p(\mathbf{z}|\tilde{\mathbf{s}}, \tilde{f}_d, \tilde{\theta}), \tag{5.17}$$

the GLRT problem can be restated as

$$\hat{\mathbf{s}}_{GLRT} = \arg \max_{\hat{\mathbf{s}}(\tilde{f}_d,\tilde{\theta}):(\tilde{f}_d,\tilde{\theta})\in\Lambda} p(\mathbf{z}|\hat{\mathbf{s}}(\tilde{f}_d,\tilde{\theta}),\tilde{f}_d,\tilde{\theta}). \tag{5.18}$$

Furthermore, collecting all candidate GLRT solutions in the sufficient set

$$\mathcal{T} \triangleq \{\hat{\mathbf{s}}(\tilde{f}_d,\tilde{\theta})|(\tilde{f}_d,\tilde{\theta})\in\Lambda\}, \tag{5.19}$$

the GLRT problem becomes

$$\hat{\mathbf{s}}_{GLRT} = \arg \max_{\tilde{\mathbf{s}}\in\mathcal{T}} p(\mathbf{z}|\tilde{\mathbf{s}},\hat{f}_d(\tilde{\mathbf{s}}),\hat{\theta}(\tilde{\mathbf{s}})). \tag{5.20}$$

It is now clear how one can proceed to prove the polynomial-complexity result: if the size of $\mathcal{T}$ grows only polynomially with N, and $\mathcal{T}$ can be constructed by a polynomial-complexity algorithm, then using (5.20), one can solve the problem with polynomial combinatorial complexity. As shown in [31] for a general GLRT problem, constructing the sufficient set $\mathcal{T}$ is equivalent to partitioning the parameter space Λ into subsets in such a way that all parameter pairs $(\tilde{f}_d,\tilde{\theta})$ in each subset result in the same sequence $\hat{\mathbf{s}}(\tilde{f}_d,\tilde{\theta})$. It was further shown that this partitioning can be accomplished by superimposing the boundaries defined by equations of the form

$$\begin{aligned} &|z_k - \sqrt{E_k}(+1)e^{j(2\pi\tilde{f}_d k+\tilde{\theta})}|^2 = |z_k - \sqrt{E_k}(-1)e^{j(2\pi\tilde{f}_d k+\tilde{\theta})}|^2 \\ \Leftrightarrow &2\pi\tilde{f}_d k + \tilde{\theta} = \angle z_k \pm \frac{\pi}{2} + 2\pi m, \end{aligned} \tag{5.21}$$

for all $k = 1, 2, \ldots, N$, and all integers m. An example of the resulting partitioned parameter space is shown in Fig. 5.1.

By observing that there are $2k$ lines for each k, we conclude that there are $O(N^2)$ lines in total implied by (5.21). Using a well-known result from computational ge-

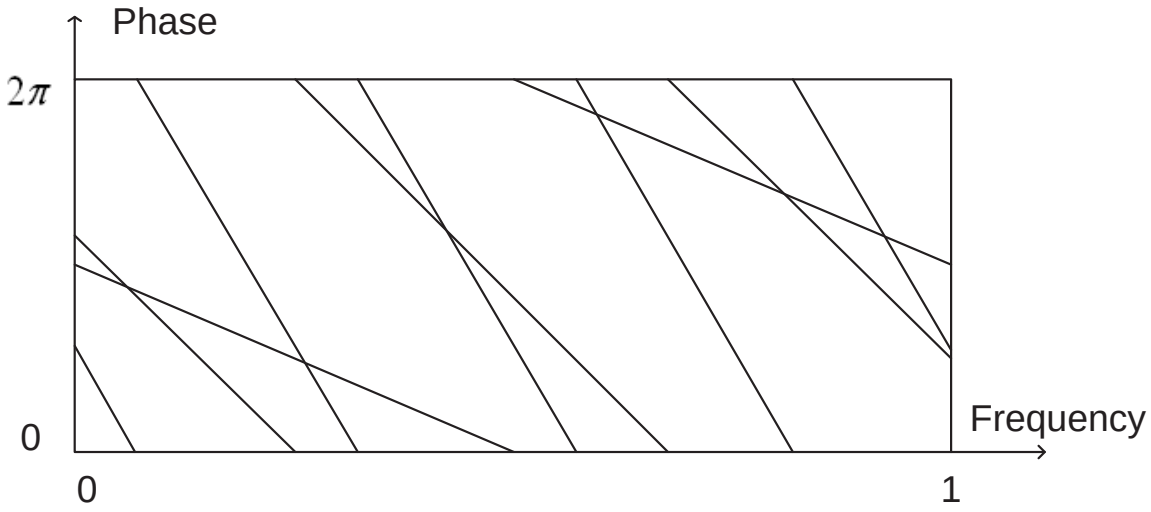


Figure 5.1: An example of the partitioned parameter space for $N = 4$.

ometry [79], we can deduce that at most $|\mathcal{T}| = O(N^4)$ polygons can be generated by superimposing $O(N^2)$ lines and that there is an $O(N^4)$-complexity algorithm that can construct these polygons, and thus the sufficient set $\mathcal{T}$. We further improve this worst-case estimate by using the additional information that every line with $k = i$ intersects at most one line with $k = j, j \neq i$, and is parallel to all lines with $k = i$. Indeed, due to the above observation, it is straightforward to show that there are at most $|\mathcal{T}| = O(N^3)$ partitions resulted from these lines since each line intersects at most $N - 1$ other lines. Moreover, to compute all intersection points of each additional line at $k = i$ with all previously drawn lines, the following algorithm can be used.

1. Find the entrance intersection point on the polygon adjacent to the left or upper boundary of the parameter space Λ. It is sufficient to look only at the left and upper boundary, because all lines have negative slopes. The complexity of this step is $O(i^2)$ for all $2i$ lines.

2. Find the exit intersection point among all sides of this polygon. Note that the number of sides per polygon is constant on the average, resulting in $O(1)$

complexity for this step.

3. Identify the next polygon ($O(1)$ complexity with proper indexing). If there is no next polygon, i.e., the exit point hits the lower or right boundary, then quit. Otherwise, go to step 2.

Since there are $O(N^2)$ lines in total, and each line visits at most N polygons as shown above, we have shown that the size of the partition and its construction complexity are both $O(N^3)$.

After constructing all $O(N^3)$ polygons, we take an arbitrary parameter pair (f_d, θ) from each polygon and do coherent symbol by symbol detection hypothesizing (f_d, θ) as the CSI. The resulting $O(N^3)$ sequences from all polygons then constitute the sufficient set $\mathcal{T}$. Finally, the frequency estimate and the noncoherent metric are evaluated for each of these sequences, and the latter is further maximized to find the GLRT solution according to

$$\hat{\mathbf{s}}_{GLRT} = \arg\max_{\tilde{\mathbf{s}}\in\mathcal{T}} \left| \sum_{k=1}^{N} z_k^* \tilde{s}_k \sqrt{E_k} e^{j2\pi \hat{f}_d(\tilde{\mathbf{s}})k} \right|. \tag{5.22}$$

Since these two final steps require $O(N)$ complexity for each polygon and we have $O(N^3)$ polygons, we have verified that the whole algorithm finds the exact GLRT solution with $O(N^4)$ complexity. We summarize the above discussion in the following proposition.

Proposition 5.1 *The proposed algorithm in this subsection finds the exact GLRT solution for model* (5.6) *with worst-case complexity* $O(N^4)$ *when omitting the complexity of the frequency estimator, where* N *is the sequence length.*

Proof: Follows from the above discussion and the framework developed in [31]. ■

5.4 Performance Analysis of Exact and Approximate Algorithms

The purpose of this section is to demonstrate that the polynomial-complexity results developed above, aside from their conceptual value, are also useful in at least two additional respects. The first is that they enable accurate performance analysis of the corresponding exact GLRT algorithms, which would otherwise be impossible or result in loose performance bounds. The second is that they imply a family of approximate algorithms that can be easily implemented and analyzed. In the following, the exact and two approximate algorithms are analyzed for the constant phase model (5.4) and one approximate algorithm is analyzed for the linear phase model (5.6).

5.4.1 Constant Phase Model: Exact GLRT Algorithm

In this subsection, we present a performance analysis for the polynomial-complexity exact GLRT algorithm for the model in (5.4). The resulting performance upper bound turns out to be extremely tight as can be seen in section 5.5, and is now given by the following proposition.

Proposition 5.2 *Under the assumptions of the model in* (5.4)*, the sequence error probability for the exact GLRT solution (which is the same as the optimal MLSD*

solution), can be upper bounded by

$$
\begin{aligned}
&P_{MLSD} \leq \\
&2\sum_{w=1}^{N-2}\int_0^{\infty} R\left(r, E_t, \frac{E_t N_0}{2}\right) Q_1\left(\frac{r|E_t - 2wE_s|}{\sqrt{2wE_tE_s(E_t - wE_s)N_0}}, \frac{r\sqrt{E_t}}{\sqrt{2wE_s(E_t - wE_s)N_0}}\right) dr \\
&+ \int_0^{\infty} R(r, E_t, \frac{E_t N_0}{2}) Q_1\left(\frac{r|E_p - (N-1)E_s|}{\sqrt{2(N-1)E_tE_sE_pN_0}}, \frac{r\sqrt{E_t}}{\sqrt{2(N-1)E_sE_pN_0}}\right) dr \\
&+ 1 - \left[1 - Q\left(\sqrt{\frac{2E_s}{N_0}}\right)\right]^{N-1},
\end{aligned}
\tag{5.23}
$$

where

$$
E_t \triangleq E_p + (N-1)E_s \tag{5.24}
$$

denotes the total signalling energy,

$$
R(r, s, \sigma^2) \triangleq \frac{r}{\sigma^2} e^{-\frac{r^2+s^2}{2\sigma^2}} I_0(\frac{rs}{\sigma^2}) \tag{5.25}
$$

is the Ricean density function with

$$
I_0(x) \triangleq \frac{1}{\pi}\int_0^{\pi} e^{x\cos\theta} d\theta \tag{5.26}
$$

being the 0th-order modified Bessel function of the first kind,

$$
Q_1(a, b) \triangleq \int_b^{\infty} x e^{\frac{-(x^2+a^2)}{2}} I_0(ax) dx \tag{5.27}
$$

is the Marcum's Q function, and

$$
Q(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\frac{t^2}{2}} dt \tag{5.28}
$$

is the Gaussian tail function.

Proof: See Appendix C.1. ■

The essence of the proof is to use union bound over a much smaller sufficient set of sequences rather than all the 2^N sequences. This can be done mainly because the set of all possible MLSD solutions had been tremendously reduced to have a size linear in N as was shown in Section 5.3.1.

5.4.2 Constant Phase Model: Pilot-Only (PO) Algorithm

It is possibly the simplest approximate algorithm, which uses the channel information provided by the pilot symbol only, and performs symbol-by-symbol detection as follows

$$\hat{s}_i = \arg\max_{\tilde{s}\in\mathcal{A}} p(z_i|\tilde{s}, z_1) \quad i = 2, \ldots, N. \tag{5.29}$$

This pilot-only (PO) algorithm has complexity $O(N)$, and its performance is given by the following proposition.

Proposition 5.3 *The probability of bit error of the PO algorithm of the model in* (5.4) *is exactly*

$$P_b(PO) = 1 - \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T\left(y + x, \frac{E_s}{N_0}\right) T\left(x, \frac{E_p}{N_0}\right) dydx, \tag{5.30}$$

where

$$T(x, S) \triangleq \frac{1}{2\pi} e^{-S} + \sqrt{\frac{S}{\pi}} \cos(x) e^{-S\sin^2(x)} \left[1 - Q(\sqrt{2S}\cos(x))\right] \tag{5.31}$$

is the probability density function (pdf) of the phase component of a circularly sym-

metric complex Gaussian random variable with mean $\sqrt{S}$ and variance 1 (see [80] for more information on $T(x, S)$). Therefore, the corresponding probability of sequence error is

$$P_{PO} = 1 - \left[\int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T\left(y + x, \frac{E_s}{N_0}\right) T\left(x, \frac{E_p}{N_0}\right) dydx \right]^{N-1}. \tag{5.32}$$

Proof: See Appendix C.2. ■

It is worth noting that even if we drop the magnitude information of z_1 and condition only on $\angle z_1$ in (5.29), we will still obtain the same performance in this uncoded sequence case. However, if we are also interested in the soft decisions of the receiver as required in the coded sequence case, then the magnitude information provided by the pilot symbol would help improve the performance as shown in [81].

5.4.3 Constant Phase Model: Uniform Sampling (US) Algorithm

Motivated by the exact algorithm described in Section 5.4.1, we now consider a simple approximate analogue of it and analyze its performance. The idea behind this algorithm is that instead of optimally partitioning the parameter space $[0, 2\pi)$–a process that has complexity $O(N \log N)$–one can partition the parameter space in a sensible, but somewhat arbitrary way, in order to reduce complexity. The simplest such partitioning may be the uniform partitioning (or sampling), which results in the following algorithm.

1. Define L samples $\{a_i\}_{i=1}^{L}$ as

$$a_i \triangleq \frac{2\pi i}{L}, \quad \forall\ i = 1, 2, \ldots, L. \tag{5.33}$$

2. Obtain L candidate sequences $\hat{\mathbf{s}}(a_i)$ for all i, and select

$$\hat{\mathbf{s}}_{US} = \arg \max_{\hat{\mathbf{s}}(a_i): i \in \{1,2,\ldots,L\}} |\mathbf{z}^H \mathbf{D} \hat{\mathbf{s}}(a_i)| \tag{5.34}$$

This algorithm has complexity $O(LN)$ and a characterization of its performance is given by the following proposition.

Proposition 5.4 *Under the assumption that $L \geq 2$, the probability of sequence error for the US algorithm can be bounded by*

$$\begin{aligned} P_{MLSD} \leq P_{US} \leq & P_{MLSD} + (N-1)(N-2) \\ & \int_0^\infty R\left(r, E_t, \frac{E_t N_0}{2}\right) \left[\int_{-\frac{\pi}{L}}^{\frac{\pi}{L}} T\left(\theta - \frac{\pi}{2}, \frac{E_s}{E_t(E_t - E_s)N_0} r^2\right) d\theta\right]^2 dr \end{aligned} \tag{5.35a}$$

$$= P_{MLSD} + O\left(\frac{N^2}{L^2} e^{-\frac{E_s}{N_0}}\right). \tag{5.35b}$$

Proof: See Appendix C.3. ■

The main idea behind the proof can again be attributed to the parameter space partitioning perspective. The event that we miss the MLSD solution is exactly the event that none of our sampling points is in the same partition with the MLSD solution. It is the probability of this event that is characterized and upper bounded in the proof.

Equation (5.35a) can be numerically evaluated to provide a measure of how far the performance of the US algorithm is from that of the MLSD algorithm. Also, the asymptotic expression in (5.35b) reveals some interesting insight. First, for high SNR, the performance of this approximate algorithm converges exponentially to that of the exact algorithm for any N and $L \geq 2$. Moreover, for any fixed SNR, if L increases slightly faster than N (e.g., as $N^{1+\epsilon}$, for an arbitrarily small positive ϵ),

then the performance of this approximate algorithm approaches that of the MLSD algorithm for large N. As will be shown in Section 5.5, this latter bound is indeed pessimistic: a small L is enough to make the performance of the US algorithm close to that of the MLSD algorithm for all SNR values of interest.

5.4.4 Linear Phase Model: 2-Dimensional Uniform Sampling (US2D) Algorithm

Motivated by the good performance of the US algorithm for the constant phase model, we propose a similar approximate algorithm for the more complicated linear phase model as follows.

1. Define $Q_f Q_\theta$ samples of frequency-phase pairs as

$$(a_i, b_j) \triangleq \left(\frac{i}{Q_f}, \frac{2\pi j}{Q_\theta} \right), \forall i = 1, \ldots, Q_f, j = 1, \ldots, Q_\theta. \tag{5.36}$$

2. Obtain $Q_f Q_\theta$ candidate sequences $\hat{\mathbf{s}}(a_i, b_j)$ and the corresponding frequency estimates $f_{i,j} = \hat{f}_d(\hat{\mathbf{s}}(a_i, b_j))$ for all i, j , and select

$$\hat{\mathbf{s}}_{US2D} = \arg \max_{\substack{\hat{\mathbf{s}}(a_i,b_j) \\ i \in \{1,2,\ldots,Q_f\} \\ j \in \{1,2,\ldots,Q_\theta\}}} \left| \sum_{k=1}^{N} z_k^* \hat{s}_k(a_i, b_j) \sqrt{E_k} e^{j2\pi f_{i,j} k} \right|, \tag{5.37}$$

   where $\hat{s}_k(\cdot, \cdot)$ denotes the kth component of the vector $\hat{\mathbf{s}}(\cdot, \cdot)$.

This algorithm has complexity $O(Q_f Q_\theta N)$ and its performance is given by the following proposition.

Proposition 5.5 *Under the assumption that $Q_f \geq 4N$ and $Q_\theta \geq 4$ and for any f_d*

and θ, the probability of sequence error for the US2D algorithm can be bounded by

$$P_{CSI} \leq P_{US2D} \leq P_{CSI} + P_{GLRT} + \left\{1 - 2\prod_{k=1}^{N}[1 - ql(k)] + \prod_{k=1}^{N}[1 - 2ql(k)]\right\} + \left\{1 - 2(1-qu)^N + (1-2qu)^N\right\} \tag{5.38a}$$

$$= P_{CSI} + P_{GLRT} + O\left(\frac{N^2}{Q_f} e^{-\frac{E_s}{N_0}}\right) + O\left(\frac{N}{Q_\theta} e^{-\frac{E_s}{N_0}}\right), \tag{5.38b}$$

where P_{CSI} and P_{GLRT} are the probability of sequence error for the perfect channel state information (CSI) and the exact GLRT algorithms, respectively, and

$$ql(k) \triangleq \int_0^{\frac{2\pi k}{Q_f}} T\left(x - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx, \tag{5.39a}$$

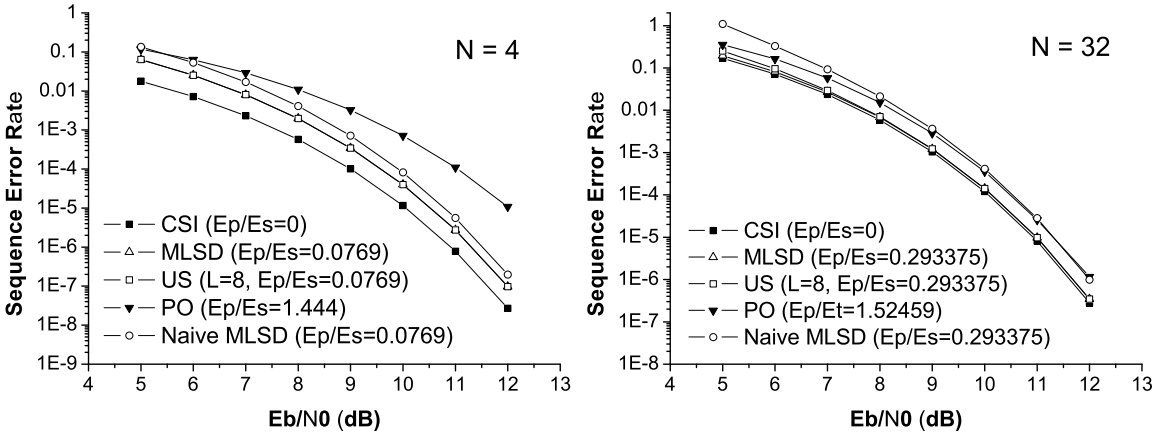
$$qu \triangleq \int_0^{\frac{2\pi}{Q_\theta}} T\left(x - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx. \tag{5.39b}$$

Proof: See Appendix C.4. ■

This result is very similar in spirit to Proposition 5.4. One difference is that because of the additional frequency jitter, we may need to increase Q_f quadratically with N in order to have a performance close to that of the exact GLRT algorithm. Another difference is that in this case, the upper and lower bounds do not agree even for large Q_f and Q_θ.

5.5 Numerical Results

In this section, numerical results are presented for the exact and approximate algorithms developed earlier for both the constant and the linear phase models.



(a) N = 4 (b) N = 32

Figure 5.2: Analytical results of exact and approximate algorithms for the constant phase model.

5.5.1 The Constant Phase Model

Fig. 5.2 compares the performance bounds for the three algorithms (where the expression for the PO algorithm is exact) developed in Section 5.4 for model (5.4) with optimally chosen pilot energies versus the information bit SNR

$$\frac{E_b}{N_0} \triangleq \frac{E_t}{N_0(N-1)\log_2|\mathcal{A}|}. \tag{5.40}$$

Also shown is the P_{CSI} performance curve, and a "naive" MLSD bound obtained by a union bound over all 2^N sequences. As can be seen in the figure, the upper bound of the US algorithm, which is obtained by substituting the MLSD bound into (5.35a), performs almost identically to the upper bound of the exact MLSD algorithm, and is very close to the performance curve of the perfect CSI receiver. With a moderate choice of $L = 8$ in the approximate algorithm, it performs equally well for both $N = 4$ and $N = 32$ cases. As compared with the PO algorithm, we gain 2 dB when $N = 4$ and about 0.5 dB when $N = 32$ by using the US algorithm. Note also that the upper bound for the MLSD algorithm is actually very tight since

the CSI performance is a lower bound.

5.5.2 The Linear Phase Model

In this subsection we provide various simulation results regarding the implementation of the US2D algorithm. For simplicity, we use $E_p = E_s$ instead of optimizing E_p throughout our simulations.

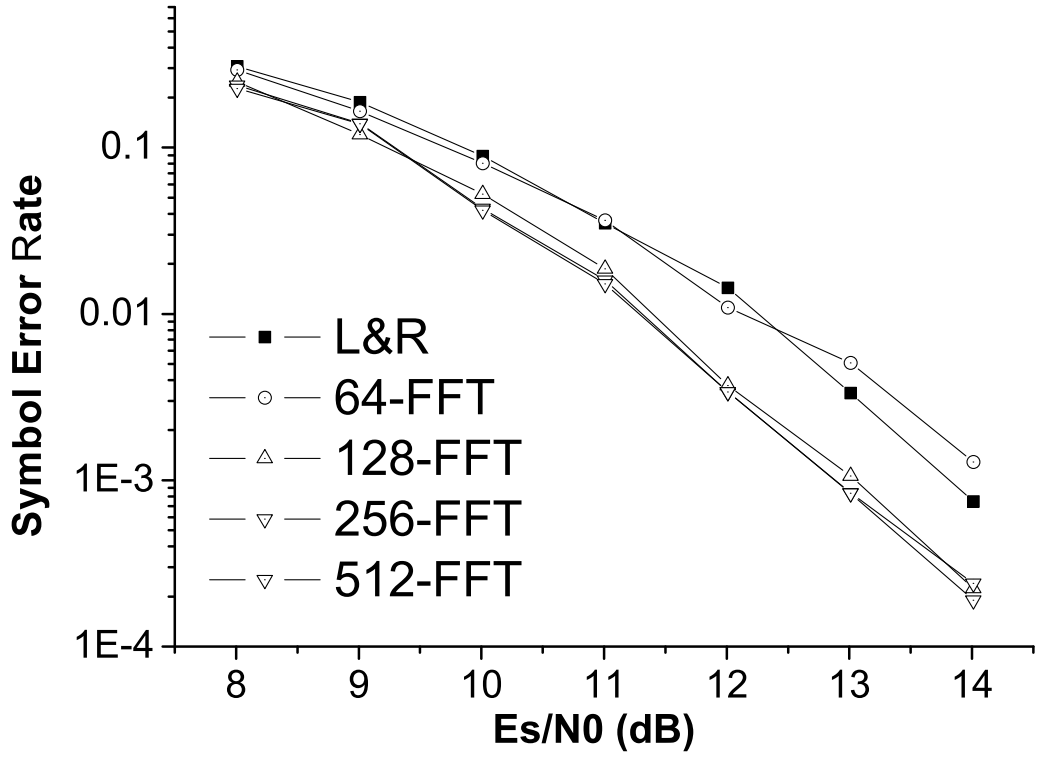


Figure 5.3: Comparison of simulation results between different frequency estimators for QPSK modulated sequences with $N = 16$, $f_d = 0.2$, for the US2D algorithm with large Q_f and $Q_\theta = 4$.

One way to perform frequency estimation for a given sequence, is to zero-pad the sequence with D zeros and perform an $(N + D)$-point FFT to find the frequency component with the maximum magnitude. This entails $O((N + D)\log(N + D))$ complexity per sequence, which can be undesirable in the US2D algorithm. Since the US2D algorithm is aiming at low complexity, we adopt the simplest version of

the Luise and Reggiannini (L&R) estimator [82] for this algorithm, which evaluates a frequency estimate as

$$\hat{f}_d(\mathbf{s}) = \angle \sum_{i=1}^{N-1} (z_{i+1} s_{i+1}^*)(z_i s_i^*)^*, \tag{5.41}$$

thus having linear complexity in N. Note that in general, the L&R estimator can use more autocorrelation coefficients than just the first one as in this case. As can be seen in Fig. 5.3, the considerably much faster L&R estimator suffers only about 1 dB performance loss (at BER=10^{-3}) compared to the higher accuracy FFT estimators. Therefore, in all the following simulations, we will employ this L&R frequency estimator for the US2D algorithm.

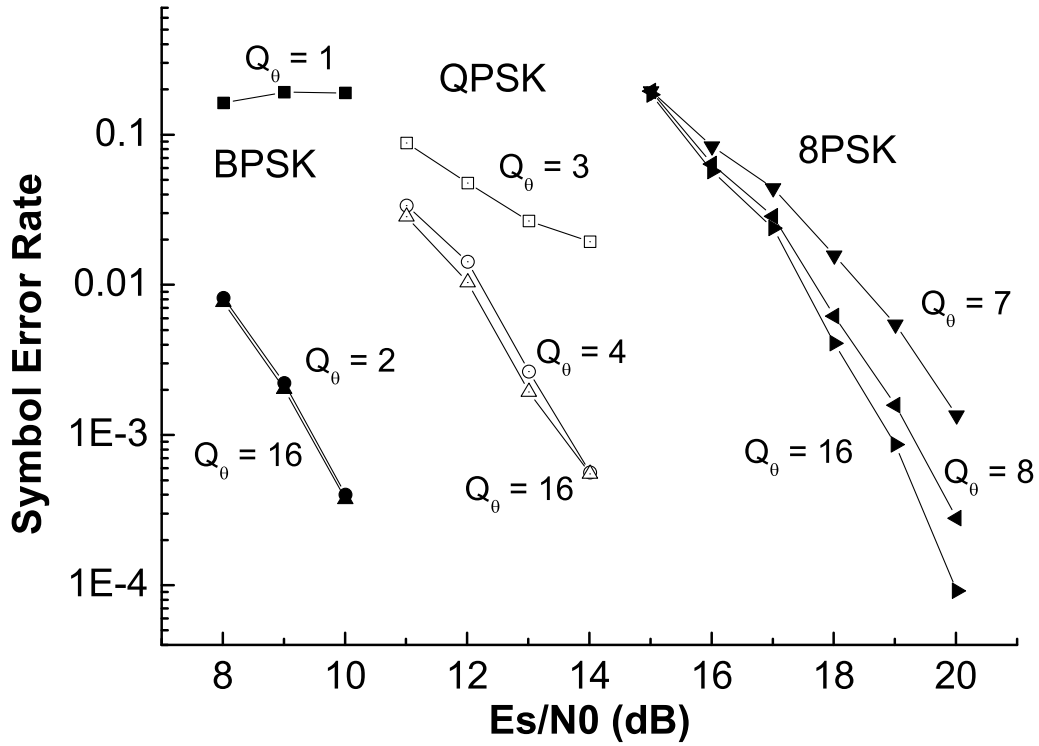


Figure 5.4: Relation between M and Q_θ for the simulated US2D algorithm with $N = 16$, $f_d = 0.2$, and a large Q_f.

To examine the effect of Q_θ in the US2D algorithm, we have nullified the effect of

Q_f by using a large Q_f such that no performance can be gained by further increasing this Q_f. Fig. 5.4 shows the performance of the US2D algorithm for a large enough Q_f, different values of Q_θ, and different M-PSK alphabets. As can be seen in the figure, very little improvement can be gained by selecting $Q_\theta > M$ regardless of N. This suggests that Proposition 5.5, which shows that performance behaves as $O(\frac{N}{Q_\theta})$, provides a conservative estimate.

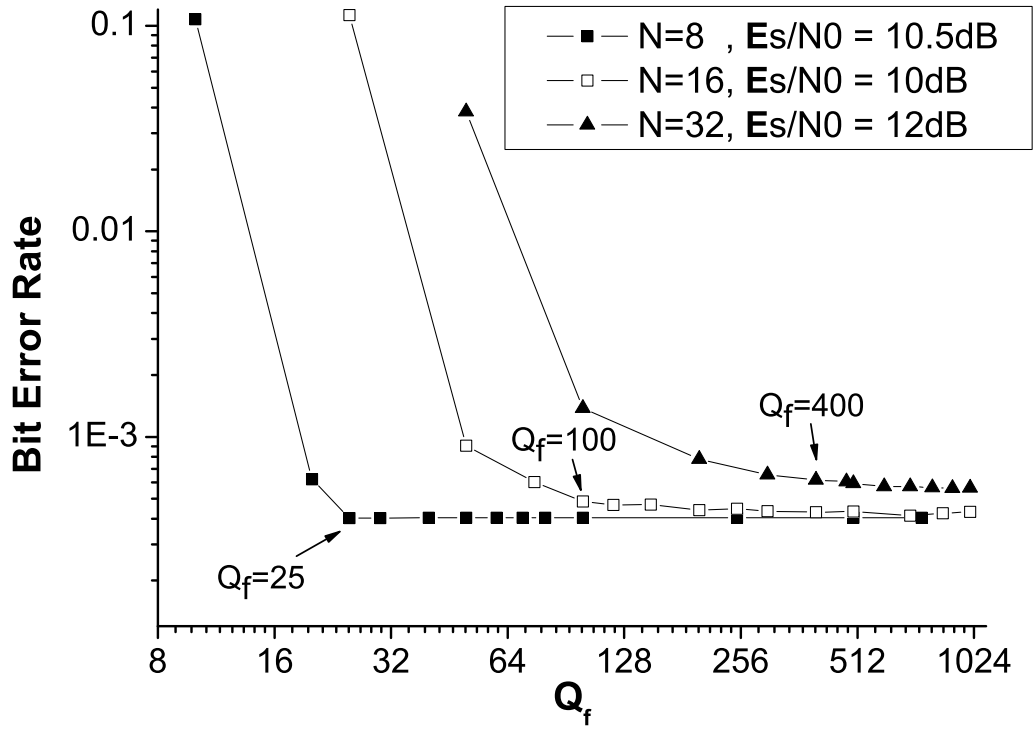


Figure 5.5: Effect of Q_f on the simulated US2D algorithm with BPSK signals and f_d, θ uniformly random. In all simulations, $Q_\theta = 2$. The different SNR values are chosen to make all performance curves saturate in approximately the same bit error rate.

Due to the above observation, we fix $Q_\theta = M$ in the US2D algorithm and investigate how Q_f affects performance. Fig. 5.5 clearly shows that to achieve the best performance, Q_f should grow approximately quadratically with N as predicted in Proposition 5.5. This result for Q_f together with the previous one for Q_θ suggests that it is possible to employ an $O(N^3)$-complexity US2D algorithm to achieve

a near-exact performance.

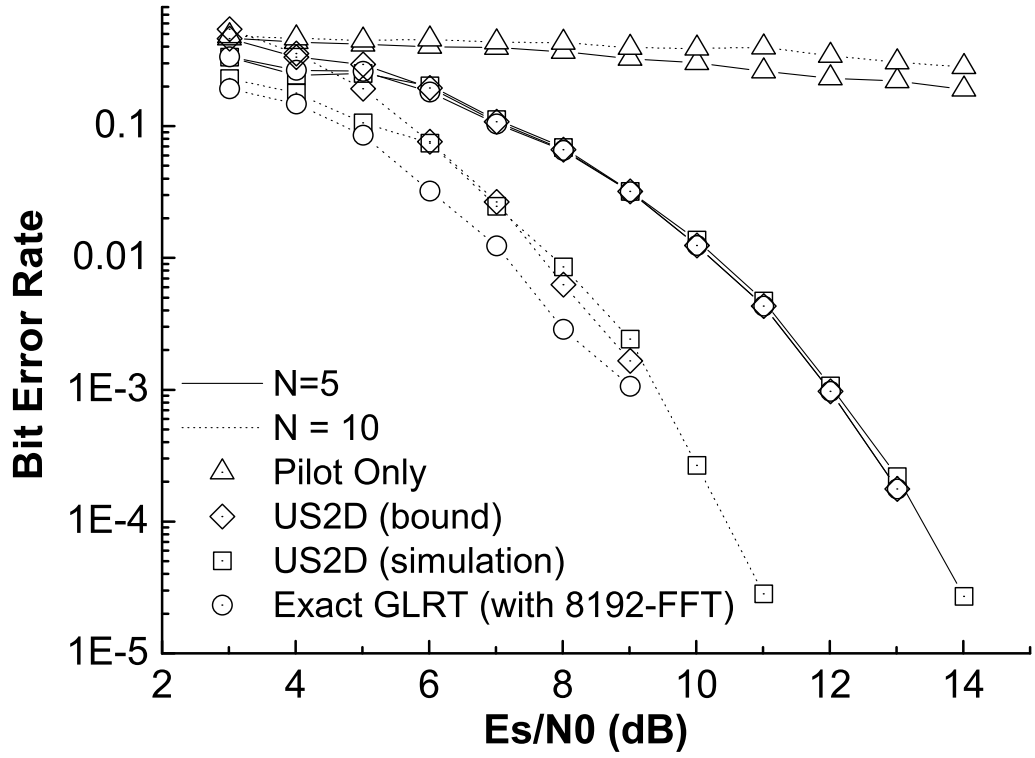


Figure 5.6: Simulation results of the exact GLRT, US2D, and an ad hoc pilot-only algorithm for BPSK modulated sequences with f_d, θ uniformly random. $Q_\theta = 2$ and $Q_f = 10$, 40 for $N = 5$, 10, respectively for the simulated US2D algorithm.

Fig. 5.6 shows that the US2D algorithm with a fast L&R frequency estimator can indeed achieve a performance close to that of the exact GLRT algorithm, which employs a very accurate FFT-based frequency estimator, by choosing a moderate Q_f and Q_θ. Notice that the number of frequency samples was increased quadratically with N. Also shown is the performance of an ad-hoc pilot-only algorithm, which first estimates f_d and θ using the first two pilots and then does symbol-by-symbol detection hypothesizing the estimated CSI. It is obvious that for this linear phase model, it is much more desirable to extract the channel information from the whole received sequence by performing joint detection/estimation schemes rather than restricting our attention only to the channel information provided by pilots. Also shown in

Fig. 5.6 are the upper bounds derived in Proposition 5.5. However, the parameters used for obtaining these bounds were $Q_\theta = 8$ and $Q_f = 8N$ which are different from those used to obtain the simulated performance. This was due to the fact that the bound is valid for $Q_\theta \geq 4$ and $Q_f \geq 4N$ as mentioned in Proposition 5.5.

5.6 Conclusion and Discussion

Several low complexity joint detection/estimation algorithms for noncoherent channels with an unknown phase rotation are presented and analyzed in this chapter. If the phase process is modelled by a constant phase shift, then we showed that the low complexity US algorithm can be applied, which yields a performance close to that of the ML receiver with perfect CSI. In the case that the phase process is modelled by both a frequency jitter and a phase shift, we showed that the exact GLRT solution can be obtained with combinatorial complexity $O(N^4)$ or well approximated by the proposed US2D algorithm with complexity $O(Q_f Q_\theta N)$.

When powerful codes are utilized, symbol by symbol soft decisions will be desired by the decoder for iterative decoding. The proposed algorithms can also be adapted to provide this information. It is very similar to the one discussed in [31], and the additional manipulation will only increase the whole complexity of the algorithms by a factor of N.

We conclude by noting that the basic idea behind the exact algorithms is to think of decision regions in the parameter space rather than in the observation space as is the traditional approach. This is helpful since in the discussed problems, the dimension of the observation space grows with N, while that of the parameter space remains fixed. It is thus expected that similar results will hold every time the number of independent parameters that need to be estimated grows slower than N.

CHAPTER 6

Polynomial Complexity Optimal Decoding of Trellis Codes Transmitted through Fading Channels

6.1 Introduction

The problem of optimal decoding of trellis coded sequences transmitted over a frequency non-selective, time-selective complex fading channel is considered in this chapter. It is a well-known fact that when the channel state information (CSI) is known to the receiver, the receiver can use Viterbi's algorithm (VA) to find the maximum a posteriori probability sequence detection (MAPSqD) solution with linear complexity in sequence length, N. However, when the CSI is not available at the receiver, the MAPSqD solution cannot be obtained using such a simple dynamic programming technique, due to the memory imposed on the received sequences by the channel. One approach to solve this problem is to transmit regularly spaced pilot symbols. The receiver can estimate the channel using the pilots and then use VA to decode the sequence based on the estimated CSI. Although this might be a satisfactory approach for high signal-to-noise ratio (SNR) applications, the unreliably

estimated CSI provided by pilots may substantially deteriorate the decoding performance when the operating SNR is low, e.g., when high-performance codes are used. In this case, joint sequence decoding and channel estimation, e.g., true MAPSqD in the presence of unknown CSI, appears to be a desirable policy.

There is an extensive set of literature on suboptimal algorithms for solving the joint decoding/estimation problem. The expectation-maximization (EM) algorithm [68, 83] performs a two-step statistical iteration between channel-conditioned sequence decoding and data-conditioned channel estimation. A family of algorithms can be constructed by viewing this problem as a hypothesis testing problem with each hypothesis (sequence) being a path in a tree of depth N. Since testing all hypotheses amounts to exponential complexity, a tree-pruning algorithm, such as the T-algorithm [69], the M-algorithm [70], or the per-survivor processing (PSP) [75] algorithm can be employed to trade off complexity for performance. In all these works, the underlying assumption was that the optimal (exact) MAPSqD solution can only be found with an exponential complexity in the sequence length N due to the exponential growth of the sequence tree.

It is our intention to prove that the exact MAPSqD solution can be obtained with only polynomial complexity in the sequence length, N. Similar problems have been addressed in the case of uncoded sequences for a class of channel models in [31, 84]. The basic idea behind these works is that although the sequence tree (and thus the number of hypotheses) grows exponentially in N, there is only a certain number of sequences that are potential candidates for the MAPSqD solution. This sufficient set of sequences is not known a-priori. But once the noisy observation is obtained at the receiver, there is a polynomial-complexity algorithm to obtain it [31, 84]. This algorithm is derived by defining a new kind of "decision regions" that partition the channel parameter space, as contrast to the traditionally defined decision

regions that partition the observation space. By studying the structure of these new "decision regions", the authors showed that their number grows only polynomially with N and that there is a polynomial-complexity algorithm that constructs them. Unfortunately, all arguments used in [31, 84] rely heavily on the assumption of uncoded-sequence transmission.

In this chapter, in order to solve the MAPSqD problem for trellis coded sequences, we adopt the concept of "decision regions" defined in the parameter space as in [31, 84]. Contrary to the previous works, however, we define a set of sufficient survivor sequences and study their evolution in time. In particular, we show that this set can be updated in a forward recursive fashion and that the cardinality of the resulting set grows only polynomially, thus establishing the polynomial-complexity result for the coded case.

The same ideas can be used to define both forward and backward sufficient survivor sets. This essentially means that the *exact* symbol-by-symbol soft decisions (more specifically, the messages corresponding to the min-sum algorithm [85]) can also be generated with polynomial complexity. Applications that can potentially benefit from this development include serially concatenated convolutional codes through flat-fading channels, where now the entire system consisting of the inner trellis code and the channel can provide an exact "soft inverse" [86] [87, p. 85] with a feasible computational complexity.

The remaining of this chapter is structured as follows. In Section 6.2, we formulate the MAPSqD problem for the trellis coded sequence transmission over flat-fading channels, and relate it to another equivalent problem. After that, Section 6.3 is devoted to reformulate this equivalent problem so as to introduce the sufficient set of the candidates of the MAPSqD solution. Based on this reformulation, an algorithm is proposed in section 6.4 to find the MAPSqD solution and also proved to require

only a polynomial complexity for the simple case where the trellis has two states. Finally, we conclude this chapter in section 6.5.

6.2 System and Channel Model

Consider the transmission of a sequence of information symbols $\mathbf{a} = (a_1, a_2, \ldots, a_N)^T$, where T denotes the transpose of some matrix, and $a_k \in \mathcal{A} \triangleq \{1, \ldots, K\}$ for all k. This sequence is encoded and modulated by a finite state machine (FSM) described by the following items.

1. Its state $s_k \in \mathcal{S} \triangleq \{1, \ldots, I\}$ at time k with the assumption that the initial state s_0 is known by both transmitter and receiver

2. The transition (or trellis edge) e_k at time k, determined by a given previous state s_{k-1} and a current input a_k, such that $e_k \triangleq (s_{k-1}, a_k) \in \mathcal{E} \triangleq \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}\}$

3. The "next-state" function $ns : \mathcal{E} \to \mathcal{S}$, such that $s_k = ns(e_k)$ for all k

4. The "output function" $out : \mathcal{E} \to \mathcal{O} \triangleq \{1, \ldots, M\}$, such that the transmitted M-ary phase shift keying (M-PSK) symbol is

$$y_k = \sqrt{E_s} e^{j\frac{2\pi}{M} out(e_k)}, \forall k \tag{6.1}$$

   with E_s being the symbol energy

5. The "previous-state function" $ps : \mathcal{E} \to \mathcal{S}$, such that $s_{k-1} = ps(e_k)$

6. The "input function" $in : \mathcal{E} \to \mathcal{A}$, such that $a_k = in(e_k)$

Suppose the sequence $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$ is transmitted through a frequency-non-selective/time-selective complex fading channel, and assume that the channel

remains constant for the entire sequence transmission. The observation model can be expressed as

$$\mathbf{z} = c\mathbf{y} + \mathbf{n}, \tag{6.2}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_N)^T$ is the received sequence, c is a zero-mean circularly symmetric complex Gaussian random variable with variance $E[|c|^2] = 1$, $\mathbf{n} = (n_1, n_2, \ldots, n_N)^T$ is a vector of independent identically-distributed (iid) zero-mean circularly symmetric complex Gaussian random variables with variance N_0.

When no CSI is available at the receiver, i.e., when the realization of c is unknown, the MAPSqD solution to this problem is

$$\hat{\mathbf{a}}_{MAPSqD} = \arg\max_{\mathbf{a}\in\mathcal{A}^N} p(\mathbf{z}|\mathbf{a})p(\mathbf{a}) = \arg\max_{\mathbf{a}\in\mathcal{A}^N} \left\{ \ln p(\mathbf{a}) + \frac{1}{N_0(N_0 + NE_s)} |\mathbf{z}^H\mathbf{y}|^2 \right\}, \tag{6.3}$$

where H denotes the Hermitian transpose of some matrix. Due to the linear and Gaussian nature of the observation model, this problem can be expressed in a double maximization form (see [88] for details) as

$$\hat{\mathbf{a}}_{MAPSqD} = \arg\max_{\mathbf{a}\in\mathcal{A}^N} \max_{c\in\mathbb{C}} \left\{ \ln p(\mathbf{a}) - \frac{1}{N_0}|\mathbf{z} - c\mathbf{y}|^2 - |c|^2 \right\}, \tag{6.4}$$

where $\mathbb{C}$ is the set of all complex numbers. Since there is a one-to-one correspondence between $\mathbf{e}$ and $\mathbf{a}$, finding $\hat{\mathbf{a}}_{MAPSqD}$ is equivalent to finding $\hat{\mathbf{e}}_{MAPSqD}$ as follows

$$\hat{\mathbf{e}}_{MAPSqD} = \arg\max_{\mathbf{e}\in\tilde{\mathcal{E}}^N} \max_{c\in\mathbb{C}} L^N(\mathbf{e}, c), \tag{6.5}$$

where

$$L^k(\mathbf{e}^k, c) \triangleq \sum_{i=1}^{k} L_i(e_i, c), \tag{6.6}$$

$$L_k(e_k, c) \triangleq \ln p(in(e_k)) - \frac{1}{N_0}|z_k - c\sqrt{E_s}e^{j\frac{2\pi}{M}out(e_k)}|^2 - \frac{|c|^2}{N}, \tag{6.7}$$

$\mathbf{e}^k \triangleq (e_1, e_2, \ldots, e_k)^T$, and

$$\tilde{\mathcal{E}}^k \triangleq \{\mathbf{e}^k : ns(e_i) = ps(e_{i+1}),\ \forall i = 1, 2, \ldots, k-1,\ ps(e_1) = s_o\}, \tag{6.8}$$

which is the set of all valid paths up to time k.

6.3 The Sufficient Set of Survivor Matrices

For a given c, if we define the survivor that ends in state i at time k obtained by the VA as

$$\hat{\mathbf{V}}^k(i|c) = \arg \max_{\mathbf{e}^k \in \tilde{\mathcal{E}}^k : ns(e_k) = i} L^k(\mathbf{e}^k, c),\ \forall i \in \mathcal{S}, \tag{6.9}$$

then we have the following lemma.

Lemma 6.1

$$\hat{\mathbf{e}}_{MAPSqD} = \hat{\mathbf{V}}^N(i|c) \textit{ for some } i \in \mathcal{S} \textit{ and } c \in \mathbb{C} \tag{6.10}$$

Proof: See Appendix D.1. ■

Therefore, if we define $\hat{\mathbf{V}}^k(c) \triangleq (\hat{\mathbf{V}}^k(1|c), \hat{\mathbf{V}}^k(2|c), \ldots, \hat{\mathbf{V}}^k(I|c))^T$ to be the survivor matrix consisting of all survivors that end in different states conditioned on parameter

c at time k, and collect all such survivor matrices in a set

$$\hat{\mathcal{D}}^k \triangleq \{\hat{\mathbf{V}}^k(c) : c \in \mathbb{C}\}, \tag{6.11}$$

then **instead of searching $\hat{\mathbf{e}}_{MAPSqD}$ in $\tilde{\mathcal{E}}^N$, we need only search through all rows of all survivor matrices in a potentially smaller sufficient set $\hat{\mathcal{D}}^N$.** However, since $\mathbb{C}$ is an infinite set, constructing $\hat{\mathcal{D}}^N$ by going through all $c \in \mathbb{C}$ requires infinite complexity. We make the following observation: it is sensible to expect that the function $\hat{\mathbf{V}}^k(c)$ remains constant for a range of c's. More rigorously, we can partition $\mathbb{C}$ in such a way that for all c in each set of the partition, $\hat{\mathbf{V}}^k(c)$ is the same. This leads us to the definition of the "parameter space decision regions"

$$T^k(\mathbf{V}^k) \triangleq \{c \in \mathbb{C} : \hat{\mathbf{V}}^k(c) = \mathbf{V}^k\}, \tag{6.12}$$

defined for every valid survivor matrix $\mathbf{V}^k \in \mathcal{D}^k$, where

$$\begin{aligned}\mathcal{D}^k \triangleq \{(\mathbf{e}^{(1)k}, \mathbf{e}^{(2)k}, \ldots, \mathbf{e}^{(I)k})^T : \mathbf{e}^{(i)k} \in \tilde{\mathcal{E}}^k, ns(e_k^{(i)}) = i,\ \forall i \in \mathcal{S},\ \text{and} \\ e_l^{(i)} \neq e_l^{(j)} \Rightarrow ns(e_l^{(i)}) \neq ns(e_l^{(j)}),\ \forall i, j \in \mathcal{S}, 1 \leq l \leq k\}\end{aligned} \tag{6.13}$$

is the set of all valid survivor matrices at time k. The constraints in (6.13) imply that once survivors merge, they have to stay merged for their entire past. With the introduction of the "parameter space decision regions", it is now clear that the sufficient set can be constructed as

$$\hat{\mathcal{D}}^k = \{\mathbf{V}^k \in \mathcal{D}^k : T^k(\mathbf{V}^k) \neq \phi\}. \tag{6.14}$$

In the next section, we show that $\hat{\mathcal{D}}^N$ and $T^N(\mathbf{V}^N)$, $\forall \mathbf{V}^N \in \hat{\mathcal{D}}^N$ can be generated with polynomial complexity in N, and furthermore, that the size of the resulting

sufficient set $\hat{\mathcal{D}}^N$ is also polynomial in N.

6.4 Recursive Construction of the Sufficient Set

Define the set of all possible extensions of $\mathbf{V}^k$ as

$$ext(\mathbf{V}^k) \triangleq \left\{\mathbf{W}^{k+1} \in \mathcal{D}^{k+1} : \mathbf{W}^{k+1}(i,k+1) = e \Rightarrow \mathbf{W}^{k+1}(i) = [\mathbf{V}^k(ps(e)), e]\right\}, \tag{6.15}$$

where we use $\mathbf{B}(i,j)$ to denote the jth element of the ith row in the matrix $\mathbf{B}$, and $\mathbf{B}(i)$ to denote the ith row of $\mathbf{B}$. Also, define the following set of channel parameters

$$P^k(\mathbf{V}^k, e) \triangleq$$
$$\left\{c \in \mathbb{C} : L^{k+1}([\mathbf{V}^k(ps(e)), e]^T, c) = \max_{\tilde{e}\in\mathcal{E}:ns(\tilde{e})=ns(e)} L^{k+1}([\mathbf{V}^k(ps(\tilde{e})), \tilde{e}]^T, c)\right\}. \tag{6.16}$$

Observe that the set $P^k(\mathbf{V}^k, e)$ is a convex polytope since its boundaries are straight lines in $\mathbb{C}$. It should be clear from this definition that for any $c \in P^k(\mathbf{V}^k, e)$, if the survivor matrix at time k is $\mathbf{V}^k$, then the VA (conditioned on c) will choose $[\mathbf{V}^k(ps(e)), e]$ as one of the survivor extensions. As a consequence of the VA, we have the following lemma.

Lemma 6.2

$$T^{k+1}(\mathbf{V}^{k+1}) = \bigcup_{\mathbf{W}^k:\mathbf{V}^{k+1}\in ext(\mathbf{W}^k)} \left\{T^k(\mathbf{W}^k) \bigcap_{i\in\mathcal{S}} P^k(\mathbf{W}^k, \mathbf{V}^{k+1}(i,k+1))\right\} \tag{6.17}$$

Proof: See Appendix D.2. ■

Lemma 6.2 essentially suggests a recursive algorithm for constructing $T^{k+1}(\mathbf{V}^{k+1})$. In addition, the set $\hat{\mathcal{D}}^{k+1}$ can be easily obtained by collecting all $\mathbf{V}^{k+1}$ such that

$T^{k+1}(\mathbf{V}^{k+1})$ is nonempty. However, since the set $P^k(\mathbf{W}^k, e)$ needs to be generated for all $\mathbf{W}^k \in \hat{\mathcal{D}}^k$, the size of $\hat{\mathcal{D}}^{k+1}$ can grow (in the worst case) as $\beta|\hat{\mathcal{D}}^k|$ for some $\beta > 1$, and thus the size of the sufficient set $\hat{\mathcal{D}}^N$ will be exponential in N.

To overcome this problem, we modify the above algorithm by utilizing the possibility that several survivor matrices $\mathbf{W}^k$ may result in the same set $P^k(\mathbf{W}^k, e)$. In particular, if $\mathbf{W}^k$ and $\mathbf{U}^k$ are such that

$$\mathbf{W}^k(i,j) \neq \mathbf{U}^k(i,j) \Rightarrow \mathbf{W}^k(i,j) = \mathbf{W}^k(l,j) \text{ and } \mathbf{U}^k(i,j) = \mathbf{U}^k(l,j)\ \forall i, l \in \mathcal{S}, \tag{6.18}$$

i.e., if they only differ in positions at which all the survivors merge together, then $P^k(\mathbf{W}^k, e) = P^k(\mathbf{U}^k, e)$, $\forall e \in \mathcal{E}$. Therefore, we can partition $\hat{\mathcal{D}}^k$ into groups $G_1^k, G_2^k, \ldots, G_{\alpha_k}^k$ such that any two survivor matrices that satisfy (6.18) belong to the same group. Fig. 6.1 shows an example of this partitioning.

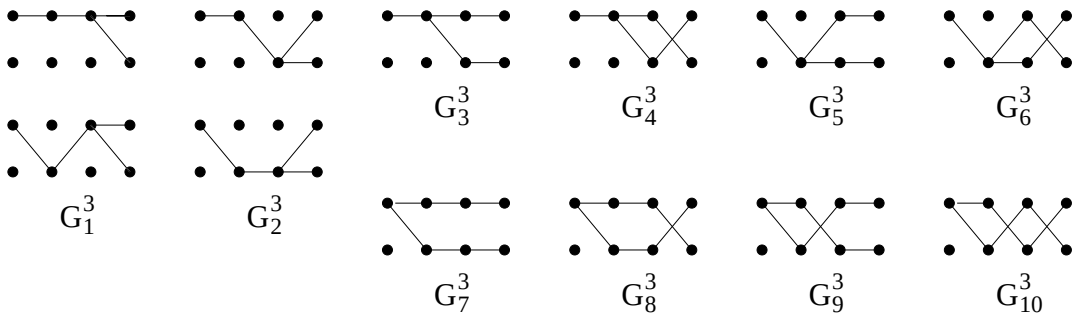


Figure 6.1: An example of groups for $I = K = M = 2$ and $k = 3$. Although in this example only G_1^3 and G_2^3 have more than one element, for larger k the sets G_i^k have a large number of elements.

With this modification, instead of constructing $P^k(\mathbf{W}^k, e)$ for all $\mathbf{W}^k \in \hat{\mathcal{D}}^k$, we can construct $P^k(\mathbf{W}^k, e)$ for only one $\mathbf{W}^k \in G_i^k$ in each group $i = 1, 2, \ldots, \alpha_k$, thus reduce complexity. This leads to the following modified algorithm:

1. Construct $T^1(\mathbf{V}^1)$ for all $\mathbf{V}^1 \in \mathcal{D}^1$. If $T^1(\mathbf{V}^1) \neq \phi$, put $\mathbf{V}^1$ into $\hat{\mathcal{D}}^1$.

2. Given $\hat{\mathcal{D}}^k$ and $T^k(\mathbf{V}^k)$ for all $\mathbf{V}^k \in \hat{\mathcal{D}}^k$, construct groups $G_1^k, G_2^k, \ldots, G_{\alpha_k}^k$.

3. For each $i = 1, 2, \ldots, \alpha_k$, do the following.

   (a) Choose an arbitrary matrix $\mathbf{W}^k \in G_i^k$, and construct $P^k(\mathbf{W}^k, e), \forall e \in \mathcal{E}$.

   (b) For all $\mathbf{V}^{k+1} = ext(\mathbf{U}^k), \mathbf{U}^k \in G_i^k$, if

$$T^k(\mathbf{U}^k) \bigcap_{i \in \mathcal{S}} P^k(\mathbf{W}^k, \mathbf{V}^{k+1}(i, k+1)) \neq \phi, \tag{6.19}$$

   then

   i. Union this set with $T^{k+1}(V^{k+1})$, where $T^{k+1}(V^{k+1})$ is initially set to be ϕ. (Note that $T^{k+1}(V^{k+1})$ is exactly what was defined in Lemma 6.2)

   ii. Put V^{k+1} into $\hat{\mathcal{D}}^{k+1}$.

4. Iteratively do steps 2) and 3) until we get $\hat{\mathcal{D}}^N$ and $T^N(\mathbf{V}^N)$.

Unfortunately, the proof of the polynomial complexity of the algorithm is only available for the case of $I = 2$. Although this is the simplest case, it has important applications, such as a differentially encoded BPSK system, and the IRA codes [17], which can be decomposed as a concatenation of an outer LDGM code and an inner accumulator code as shown in Chapter 3.

Lemma 6.3 *For $I = 2$, the number of groups α_k is at most polynomial in k.*

Proof: See Appendix D.3. ■

We conclude the main result of this chapter in the following theorem.

Theorem 6.1 *For $I = 2$, the algorithm stated in Lemma 6.2 and modified using the groups defined in* (6.18) *can find the exact $\hat{\mathbf{e}}_{MAPSqD}$ solution with worst-case polynomial complexity in N for any signal-to-noise ratio.*

Proof: Since Lemma 6.3 shows that the number of groups α_k is at most polynomial in k, it in turn implies that all sets $T^{k+1}(\mathbf{V}^{k+1})$ are polytopes defined by a number of equations which is polynomial in N. Since each equation represents a line in the complex plane, the problem becomes equivalent to finding all partitions generated by a polynomial number of lines in $\mathbb{C}$. This problem has been studied before in [79], where it is shown that there exists a polynomial-complexity algorithm that can find the at-most-polynomial number of such polytopes. The existence of this algorithm completes the proof. ■

6.5 Discussion and Conclusion

In this chapter, the problem of optimal MAPSqD of a trellis coded data sequence transmitted over a frequency-nonselective/time-selective complexity fading channel is considered. The case when the receiver does not have CSI is addressed. It is shown that, contrary to the traditional belief, the exact solution can be obtained with polynomial complexity in the sequence length (however, only the proof for the two-state trellis case is presented). The novel approach we used here to establish these results is to view this detection problem from the channel parameter space as opposed to the observation space and define appropriate decision regions.

We would like to point out that with a small modification, one can also solve the more interesting problem of obtaining symbol-by-symbol soft decisions with polynomial complexity. In particular, the metric

$$\begin{aligned} SbS_k(a) &\triangleq \max_{\mathbf{a}:a_k=a} \ln\{p(\mathbf{z}|\mathbf{a})p(\mathbf{a})\} \\ &= \max_{\mathbf{a}:a_k=a} \left\{ \ln p(\mathbf{a}) + \frac{1}{N_0(N_0+NE_s)} |\mathbf{z}^H\mathbf{y}|^2 \right\} \\ &= \max_{\mathbf{e}:in(e_k)=a} \left\{ \max_{c\in\mathbb{C}} L^N(\mathbf{e},c) \right\}, \end{aligned} \tag{6.20}$$

which is exactly the metric implied by the min-sum algorithm [8] can be obtained using the following idea. Recall that in the proposed algorithm, we have the sufficient sets $\hat{\mathcal{D}}^k$ of forward survivor matrices for all time instants k. Similarly, we can construct the sufficient sets $\hat{\mathcal{B}}^k$ of backward survivor matrices at all time instants. Therefore for any given edge e_k we can always find its best past and future evolution and the corresponding soft metric by comparing all the possible combinations of all the past and future survivors in the forward $\hat{\mathcal{D}}^{k-1}$ and backward $\hat{\mathcal{B}}^{k+1}$ sufficient sets, respectively.

CHAPTER 7

Summary and Future Directions

In this chapter, we summarize our main results presented in this thesis and discuss future research directions.

7.1 Summary

In this thesis, we consider the general problem of channel coding in digital communication systems with low computational complexity under two scenarios. In the case where the channel is a memoryless binary-input output-symmetric (MBIOS) channel, we work on design and analysis of capacity-achieving binary linear codes defined on factor graphs. On the other hand, when the channel has memory, we explore the possibility of optimal detection and decoding for uncoded and coded sequences, respectively, with a much preferable polynomial complexity to the traditionally believed exponential complexity.

In Chapter 2, we focus on the design of capacity-achieving codes through puncturing. In particular, we derive and analyze upper bounds on the average weight distribution (AWD) and asymptotic AWD (AAWD) of punctured Gallager's (n, j, k) low-density parity-check (LDPC) codes, and prove that they can achieve capacity on

any MBIOS channel using maximum-likelihood (ML) decoding if they are punctured from some original code with small enough rate and $k > j \geq 5$. Furthermore, we show that any desired rate can be achieved through puncturing without rate reduction (with respect to the original codeword length) if the original code has a small enough rate, a condition that also implies that punctured codes are "good", i.e., they have linearly increasing minimum distance, with asymptotically high probability.

Amongst all the capacity-achieving codes, the ones whose graphical representation has a bounded graphical complexity, i.e., the number of edges per information bit in the graph is bounded, are studied in Chapter 3. In particular, their ML decoding performance on MBIOS channels and belief-propagation (BP) decoding performance on the binary erasure channel (BEC) are investigated. We derive upper bounds on the AWD and AAWD of systematic and nonsystematic irregular repeat accumulate (IRA) codes, which are known to be capacity-achieving with bounded complexity on the BEC, by viewing them as serially concatenated codes with an outer low-density generator matrix (LDGM) code and an inner accumulator code. These upper bounds can be used in a variety of ML decoding performance bounds to determine the asymptotic capability of the codes. As an example, we invoke Divsalar's bound and show that nonsystematic IRA codes are guaranteed to have a better ML decoding performance than systematic IRA and LDPC codes with the same graphical complexity on the binary-input additive white Gaussian noise (BI-AWGN) channel. By numerically plotting the AAWD of IRA codes, we also see that nonsystematic IRA codes have a more concentrated asymptotic spectrum than the systematic ones, and that the inner accumulator code plays an important role in eliminating low-weight codewords.

Motivated by the open problem of whether there exist capacity-achieving codes on general MBIOS channels with bounded graphical complexity, we introduce a new

family of serially concatenated codes with an outer LDPC code and an inner LDGM code, called LDPC-GM codes. By deriving and analyzing an upper bound on the AAWD of these codes, we prove that they can achieve capacity on any MBIOS channel using ML decoding with bounded graphical complexity if the outer code is a Gallager's LDPC code and the inner code is a regular rate-1 LDGM code. Moreover, we show that they have linearly increasing minimum distance that achieves the Gilbert-Varshamov bound of all rates. These results should be attributed to the presence of the inner rate-1 LDGM code, which is shown to help eliminate high-weight codewords while maintaining a vanishingly small amount of low-weight codewords in the outer LDPC code. To investigate the BP decoding performance of these LDPC-GM codes on the BEC, we utilize the powerful density evolution (DE) method, and give two specific capacity-achieving LDPC-GM ensembles (with the outer LDPC codes allowed to be irregular) using BP decoding based on the mathematical results given in [18]. Furthermore, using the notion of puncturing, we show that LDPC-GM codes can achieve capacity on the BEC with any erasure probability if the inner LDGM code is also allowed to be irregular. The aforementioned results suggest high potential of the LDPC-GM codes for achieving capacity on MBIOS channels *using BP decoding* with bounded decoding complexity per iteration.

All results thus far assume either ML decoding on MBIOS channels or iterative decoding on the much simpler BEC. In Chapter 4 we make an attempt to bridge this gap, by analyzing the performance of iterative decoding on MBIOS channels. In particular, we derive bounds on the asymptotic performance of LDPC codes with the min-sum (MS) and BP decoding on MBIOS channels. For MS decoding, by upper bounding the bit error probability of the root bit of a tree code by the sequence error probability of a subcode of the tree code, and using the union bound, we obtain a recursive upper bound on the bit error probability after each iteration. As for the

BP decoding, we track the evolution of the probability of error and Bhattacharyya parameter associated with the outgoing message of bits after each iteration. As a result, a recursive lower bound on the probability of bit error and an upper bound on the Bhattacharyya parameter, which can also serve as an upper bound on the probability of bit error, are then derived for each iteration. Both recursions recover the one-dimensional DE equation for LDPC codes on the BEC with inequalities becoming exact equalities. This fact further implies that the asymptotic BP decoding performance of LDPC codes on the BEC is the best among all MBIOS channels with the same probability of uncoded bit error, and is the best among all MBIOS channels with the same Bhattacharyya parameter. Since this relationship between the BEC and all MBIOS channels is based purely on Fact 1.1 and Fact 1.2, it holds for the more general multi-edge type LDPC codes as well.

We turn our attention to channels with memory in Chapter 5, where we consider issues regarding the detection of uncoded sequences on additive white Gaussian noise (AWGN) channels subject also to unknown phase rotation and frequency jitter. Making the block-independent assumption and assuming the detection is based on generalized-likelihood ratio test (GLRT), we give an algorithm that finds the exact solution with $O(N^4)$ complexity, where N is the sequence length. This result is based on the parameter-space-partitioning structure introduced in [31], which is further utilized in the analysis for a family of exact and approximate algorithms in that chapter. These analytical results are then compared with various simulation results, where we show that the proposed uniform-sampling (US) algorithm and 2-dimensional uniform-sampling (US2D) algorithm well approximate the corresponding exact algorithms with much reduced complexity, and outperform the simple pilot-only (PO) algorithm.

Although the parameter-space-partitioning structure introduced in [31] and used

in Chapter 5 highly relies on the *uncoded* assumption of the transmitted sequences, we still manage to prove its feasibility for coded sequences in Chapter 6. In that chapter, we devise an algorithm that finds the optimal maximum a posteriori probability sequence detection (MAPSqD) solution for 2-state-trellis coded sequences on the frequency nonselective/time-selective fading channel with polynomial complexity. Important applications of this result, though only for 2-state-trellis coded sequences, include the differentially encoded BPSK system and the IRA codes [17], which can be decomposed into an outer LDGM code and an inner accumulator code as mentioned before.

7.2 Future Directions

Some extensions to the presented results in this thesis and directions for future research are discussed below.

7.2.1 Extensions to the Min-Sum Decoding Performance Analysis

In Chapter 4, we proved that for the (λ, ρ) LDPC ensemble with BP decoding on the MBIOS channel with Bhattacharyya parameter D, if $x_0 = D$, and

$$x_l = D\lambda(1 - \rho(1 - x_{l-1})), \quad \forall l \geq 1, \tag{7.1}$$

then asymptotically (as the codeword length goes to infinity) the average probability of bit error P_l after l decoding iterations of the LDPC ensemble satisfies

$$P_l \leq x_l, \quad \forall l \geq 0. \tag{7.2}$$

We conjecture that the same recursively determined upper bound (7.1) also holds for the MS decoding performance. As an example, Tables 7.1 and 7.2 compare the true decoding threshold and the threshold implied by the conjectured bound (7.1) for regular LDPC codes with MS decoding on the BIAWGN channel and binary symmetric channel (BSC), and show no contradiction to this conjecture.

Table 7.1: Thresholds of the variance on the BIAWGN channel for regular LDPC ensembles with MS decoding.

d_v	d_c	$\sigma_{min-sum}$	σ_{bound}
3	6	0.8177	0.7691
4	8	0.7455	0.7222
5	10	0.6957	0.6822
3	5	0.9154	0.8713
4	6	0.8716	0.8568
3	4	1.1020	1.0724
4	10	0.6776	0.6515
3	9	0.6751	0.6292
3	12	0.6079	0.5664

Table 7.2: Thresholds of the crossover probability on the BSC for regular LDPC ensembles with MS decoding.

d_v	d_c	$p_{min-sum}$	p_{bound}
3	6	0.070	0.048
4	8	0.054	0.038
5	10	0.042	0.031
3	5	0.094	0.072
4	6	0.079	0.069
3	4	0.131	0.119

Moreover, we found that we can connect the recursion (7.1) to the reduced codebook $\mathcal{C}_r$ defined in Definition 4.1, which is used in the MS decoding performance analysis in Chapter 4, as follows. Without loss of generality, let us consider the tree code $\mathcal{C}$ of $l+1$ levels associated with the regular (x^{d_v-1}, x^{d_c-1}) LDPC ensemble, and let $\mathcal{C}_r$ be the reduced codebook of $\mathcal{C}$. Define sequentially

(i) $a_{ij}^{(l)}$ = number of combinations of different i nonzero codewords in $\mathcal{C}_r$ such that the Hamming weight of the OR of these i codewords is j,

(ii) $A_i^{(l)}(x) = \sum_{j=1}^{\infty} a_{ij}^{(l)} x^j$, and

(iii) $N_l(x) = \sum_{i=1}^{\infty} (-1)^{i-1} A_i^{(l)}(x)$.

In other words, $A_1^{(l)}(x)$ is the traditional weight enumerator of the nonzero codewords, and $A_i^{(l)}(x)$ is the weight enumerator of the OR of i different nonzero codewords for all $i \geq 2$ in $\mathcal{C}_r$. We have the following lemma describing the evolution of $N_l(x)$ for each l.

Lemma 7.1 *$N_l(x) = x[1 - (1 - N_{l-1}(x))^{d_c-1}]^{d_v-1}$ for all $l \geq 0$.*

Proof: See Appendix E. ■

Therefore, if one can prove that

$$Q_l \leq N_l(D) \tag{7.3}$$

where Q_l is the probability of sequence error for C_r under maximum-likelihood sequence detection (MLSqD) on the MBIOS channel with Bhattacharyya parameter D assuming the transmission of the all-zero codeword, then it follows directly from the discussion of the reduced codebook in Chapter 4 that the recursion (7.1) gives a true upper bound on the probability of bit error for LDPC codes under MS decoding. Recall that in Chapter 4, we proved that

$$Q_l \leq A_1^{(l)}(D) \tag{7.4}$$

using the union bound. It is thus possible that by taking care of the overlaps in the

pairwise decision regions,

$$Q_l \leq N_l(D) = A_1^{(l)}(D) - A_2^{(l)}(D) + A_3^{(l)}(D) - \dots \tag{7.5}$$

can be shown to be true. However, proving (7.5) turns out to be a difficult task.

7.2.2 Graph Reduction

In Chapter 3, we proved that codes with state nodes can achieve capacity on any MBIOS channel under ML decoding with bounded graphical complexity. A practically more interesting question would be whether by introducing state nodes into the graph, finite-length codes can also have a better complexity-performance tradeoff under ML or even BP decoding. As an initial step to understanding this problem, we introduce a simple approach of adding state nodes to the graph, that can decrease the graphical complexity of any given code, maintain its ML decoding performance, and possibly increase its BP decoding performance on MBIOS channels in the following.

Let H be the $m \times n$ parity-check matrix of some given code C dictated by its graphical representation with no state nodes, i.e., an edge in the graph between the ith check node and jth variable node is reflected by a 1 in the ith row and jth column of H. Consider the following modification of the H matrix (and equivalently the graphical representation of the code):

1. Add a new check node and state node to the graph, and define the new generalized parity-check matrix H'' with state nodes of the new code C' as follows.

$$H'' = \begin{pmatrix} H & \mathbf{0}^T \\ \mathbf{x} & 1 \end{pmatrix}, \tag{7.6}$$

where $\mathbf{x}$ is some nonzero row vector of length n, $\mathbf{0}$ is the all-0 row vector of length m, and the superscript T denotes transposition of some matrix.

2. Add the last row of H'' to the other rows which have 1's in places where $\mathbf{x}$ has 1's thus resulting in the matrix H'.

Consider the code

$$C' = \{\mathbf{c} : \exists v \text{ s.t. } (\mathbf{c} \quad v)H'^T = \mathbf{0}\} \tag{7.7}$$

The resulting code C' satisfies the following lemmas.

Lemma 7.2 *$C' = C$.*

Proof: In the first step of the modification, since the additional check node only restricts the additional state node to be determined by the other variable nodes and does not interfere with the existing parity-check equations, the codebook remains unchanged. Moreover, in the second step of the modification, since only row operations are performed on H'', the codebook still remains unchanged. ■

This lemma shows that the ML decoding performance the code is unchanged on MBIOS channels by the modification. In the following lemma, we show that this modification may even increase the BP decoding performance of the code on the BEC.

Lemma 7.3 *The BP decoding performance of H' is better than or equal to that of H on the BEC.*

Proof: Let c be the additional check node and v the additional state node. It suffices to prove that if $\mathcal{S}'$ is a stopping set[1] of H', then $\mathcal{S} \triangleq \mathcal{S}' \setminus \{v\}$ is a stopping

[1]A stopping set is a set of variable nodes in the graph such that all its check node neighbors are connected to this set at least twice. It is shown in [89] that the stopping sets are exactly the sets of variable nodes in the graph that if erased, can not be recovered by the BP decoding on the BEC.

set of H. Let $\mathcal{Z}_0$ be the set of check nodes that are not connected to v on the graph H', i.e., the set of check nodes in H whose connection is not changed in the proposed modification. Also, let $\mathcal{Z}_1$ be the set of check nodes, except c, that are connected to v on H', i.e., the set of check nodes in H whose connection is changed in the proposed modification. Moreover, let $\mathcal{Z}'_s$ and $\mathcal{Z}_s$ be the sets of check nodes that are connected to some variable node in $\mathcal{S}'$ and $\mathcal{S}$ on graphs H' and H, respectively. Assuming $\mathcal{S}'$ is a stopping set of H', we want to prove that all check nodes in $\mathcal{Z}_s$ are connected to at least two variable nodes in $\mathcal{S}$.

Let z be a check node in $\mathcal{Z}_s \bigcap \mathcal{Z}_0$. Since the connection of z is not changed in the modification, and $\mathcal{S} \subset \mathcal{S}'$, we have $z \in \mathcal{Z}'_s$. Since $\mathcal{S}'$ is a stopping set, and z is not connected to v on H', it is connected to at least two variable nodes in $\mathcal{S}' \setminus \{v\} = \mathcal{S}$ on H'. Again, since the connection of z is not changed in the modification, it is connected to at least two variable nodes in $\mathcal{S}$ on H. On the other hand, when z is a check node in $\mathcal{Z}_s \bigcap \mathcal{Z}_1$, we would like to proceed by discussing three cases.

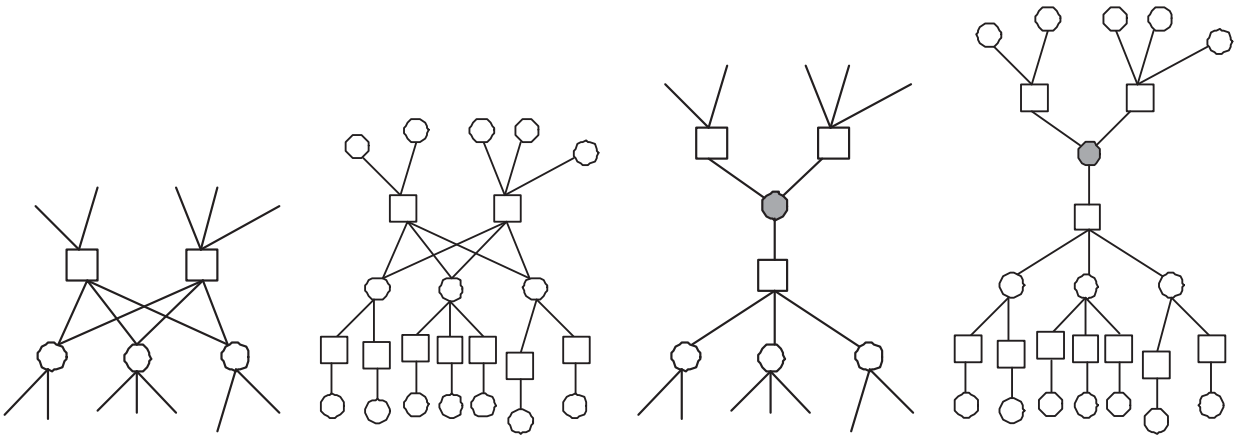
Case 1: z is not in $\mathcal{Z}'_s$. Then we know that $\mathcal{S}' \subset \mathcal{V}_1$, where $\mathcal{V}_1$ denotes the set of variable nodes, except v, that are connected to c on H', and that $\mathcal{S}' = \mathcal{S}$. Since $\mathcal{S}'$ is a stopping set of H', we have $|\mathcal{S}'| \geq 2$, where $|\cdot|$ denotes the cardinality of some set. Now, since $z \in \mathcal{Z}_1$, we have that z is connected to all the variables in $\mathcal{V}_1 \supset \mathcal{S}' = \mathcal{S}$ on H. So, z is connected to at least two variable nodes in $\mathcal{S}$ on H.

Case 2: z is in $\mathcal{Z}'_s$, but v is not in $\mathcal{S}'$. Then we have $\mathcal{S}' = \mathcal{S}$. Since, $\mathcal{S}'$ is a stopping set, z is connected to at least 2 variable nodes in $\mathcal{S}'$ on H'. With the additional connections from z to the variable nodes in $\mathcal{V}_1$, z is still connected to at least two variable nodes in $\mathcal{S}$ on H.

Case 3: z is in $\mathcal{Z}'_s$, and v is in $\mathcal{S}'$. Since v is in $\mathcal{S}'$, c is in $\mathcal{Z}'_s$. Moreover, since $\mathcal{S}'$ is a stopping set, $|\mathcal{V}_1 \bigcap \mathcal{S}| \geq 1$. Also, since z is in $\mathcal{Z}'_s$, $\mathcal{S}'$ is a stopping set, and z is not connected to any variable nodes in $\mathcal{V}_1$ on H', z is connected to at least one

variable node in $\mathcal{S} \setminus \mathcal{V}_1$ on H'. Hence, with the additional connections from z to the variable nodes in $\mathcal{V}_1$, z is connected to at least two variable nodes in $\mathcal{S}$ on H. ■

A similar result to Lemma 7.3 on general MBIOS channels can be argued as follows. Let c, v, $\mathcal{Z}_0$, $\mathcal{Z}_1$ and $\mathcal{V}_1$ be as defined in the proof of Lemma 7.3, and let $\mathcal{V}_0$ be the set of variable nodes that are not connected to c in H'. More over, let $\mathcal{E}_o$ be the set of edges connected from $\mathcal{V}_0$ to $\mathcal{Z}_1$ and from $\mathcal{V}_1$ to $\mathcal{Z}_0$, and $\mathcal{E}_i$ the set of edges connected from $\mathcal{V}_1$ to $\mathcal{Z}_1$. Then, we can view the local structure $(\mathcal{V}_1, \mathcal{Z}_1, \mathcal{E}_i)$ as a subsystem of H that outputs messages on any edge $e \in \mathcal{E}_o$ according to the input messages on the edges in $\mathcal{E}_o \setminus \{e\}$ and the channel observations of the variables in $\mathcal{V}_1$. We can as well view the input messages on $\mathcal{E}_o$ as reliability messages from channel of some imaginary variables. For the edge $e \in \mathcal{E}_o$ such that e is connected to some $v_1 \in \mathcal{V}_1$, the corresponding imaginary variable of the input message on e is connected to v_1 through a repetition code structure, i.e., through an imaginary check node connecting it and v_1. On the other hand, for the edge $e \in \mathcal{E}_o$ such that e is connected to some $z_1 \in \mathcal{Z}_1$, the corresponding imaginary variable of the input message on e is connected directly to z_1. Let H_1 be the new graph including the imaginary variable and check nodes and the original local structure. What the subsystem $(\mathcal{V}_1, \mathcal{Z}_1, \mathcal{E}_i)$ does on H_1 is some sort of extrinsic bit decoding for each imaginary variable node given the observations of all the other variable nodes in H_1. Now, let H_1' be the corresponding graph with imaginary variable and check nodes for the modified local structure $(\mathcal{V}_1 \bigcup \{v\}, \mathcal{Z}_1 \bigcup \{c\}, \mathcal{E}_i')$, where $\mathcal{E}_i'$ is modified from $\mathcal{E}_i$ as proposed. Then, since H_1' is a tree code, the modified local structure performs the exact extrinsic maximum a-posteriori probability (MAP) decoding for the imaginary variable nodes. Due to the optimality of the MAP decoding performance, we conclude that the modified local structure in H' always gives an extrinsic bit decoding performance, which is better than or equal to that given by the original local structure. See Fig 7.1



(a) Original local structure in H (b) H_1 (c) Modified local structure in H' (d) H_1'

Figure 7.1: An example of the proposed modification on a local structure, where code nodes are denoted by open circles, state nodes by filled gray circles, and check nodes by squares.

for an illustration.

It is now clear that the proposed modification can maintain the ML decoding performance and possibly increase the BP decoding performance of codes on MBIOS channels. In the following example, we show that the proposed modification can also decrease the graphical complexity for some H if we choose $\mathbf{x}$ wisely.

Example 7.1

Let

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{7.8}$$

$$\mathbf{x} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{7.9}$$

Then we have

$$H' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{7.10}$$

which is a tree code while H has loops. This example shows that H' can be better than H in terms of BP decoding performance on MBIOS channels. Moreover, comparing the number of 1's in H and H', we see that the graphical complexity is also reduced. ■

More challenging questions in this direction include whether there exist other modification algorithms that can give better complexity-performance tradeoffs, and whether there exists a modification algorithm that can find the simplest graphical representation of any given code. Advanced matrix theory may be needed in answering these questions. In addition one might consider randomized algorithms and prove existence through an averaging argument.

7.2.3 Other Research Directions

1. Let us revisit the fundamental question: "How simple can graph codes be as a function of their performance?" So far, our contribution (and the contributions in the literature) provide us with the *achievable* complexity, while the *required* complexity remains unknown for general codes with state nodes. Does this question have different answers for different decoding algorithms? Answers to these type of questions can directly impact the construction of capacity-achieving codes with the *optimal* performance-complexity tradeoff. As a first step, one can consider the simpler problem of whether a fixed proportion (with respect to the codeword length) of state nodes with a limited degree prohibits a

code with bounded graphical complexity from achieving capacity. Since every graph with state nodes is equivalent (in the sense of representing the same code) to another graph without state nodes corresponding to the parity-check matrix of the code, we may be able to solve this simpler problem by checking whether the considered graph with state nodes is equivalent to a graph without state nodes that satisfies the necessary condition for achieving capacity given by [27].

2. We show in this thesis that the decoding complexity per iteration for capacity-achieving codes can be bounded regardless of the gap to capacity on general MBIOS channels. Does this come at the price of increasing the required number of iterations? More generally, how does the required number of iterations scale as a function of the gap to capacity for capacity-achieving codes? The GEXIT chart method [62] seems to provide a framework for analysis, and seems to support the conjecture that the required number of iterations scale like $1/\epsilon$, where ϵ is the gap to capacity. However, a rigorous proof of such a statement will require further investigation into the properties of GEXIT charts.

3. We have already seen in this thesis that the rate-1 accumulator code and regular LDGM code help eliminate low-weight and high-weight codewords, respectively, for their corresponding outer codes. However, this asymptotic spectrum shaping effect is only guaranteed to help in ML performance of codes. Can a rate-1 pre-coder or post-coder also help in the iterative decoding performance? The success of the accumulate-repeat-accumulate (ARA) [54] codes seem to suggest a positive answer to this problem. For a detailed analysis, the effect of concatenating a rate-1 code on the DE equation may need to be studied.

4. Finally, although we have established that polynomial complexity optimal

MLSD can be performed for uncoded and 2-state trellis-coded sequences on channels with parametric uncertainty, the case of arbitrary trellis-coded sequences remains unanswered.

APPENDICES

APPENDIX A

Proofs of Properties of Punctured LDPC Codes in Chapter 2

A.1 Proof of Theorem 2.2

1. Since $\Delta R = 0$, we have from (2.18) that

$$w_p(0) \leq \frac{\Delta R}{1-p} = 0 \tag{A.1}$$

2. We study the behavior of $T(a,b)$ in two cases. When $[bp+(1-p)a] \in (0,\delta_o) \cup$

$(1-\delta_o, 1)$, we have from Fact 2.1 that

$$\begin{aligned}
T(a,b) =& pH(b) - H(pb+(1-p)a) + w_o(pb+(1-p)a) && \text{(A.2a)}\\
<& pH(b) - H(pb+(1-p)a) && \text{(A.2b)}\\
=& -[pb+(1-p)a]\ln\frac{b}{[pb+(1-p)a]}\\
& -[1-pb-(1-p)a]\ln\frac{1-b}{[1-pb-(1-p)a]}\\
& +(1-p)a\ln b + (1-p)(1-a)\ln(1-b) && \text{(A.2c)}\\
\le& (1-p)[a\ln b + (1-a)\ln(1-b)] && \text{(A.2d)}\\
\le& -(1-p)H(a), && \text{(A.2e)}
\end{aligned}$$

where the last two inequalities are due to the fact that relative entropy is always nonnegative [49, Theorem 2.6.3]. On the other hand, when $[bp+(1-p)a] \in [\delta_o, 1-\delta_o]$, we have from Lemma 2.1

$$\begin{aligned}
T(a,b) \le& pH(b) + (1-R_o)\ln\frac{1+[1-2(pb+(1-p)a)]^k}{2}\\
\le& p\ln 2 + (1-R_o)\ln\frac{1+[1-2(pb+(1-p)a)]^k}{2}\\
\overset{(a)}{\le}& p\ln 2 + (1-R_o)\ln\frac{1+(1-2\delta_o)^k}{2}\\
=& (1-R_o)\left[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)\right], && \text{(A.3)}
\end{aligned}$$

where (a) is true since $\ln[1+(1-2x)^k]$ is a monotonically decreasing function in x, which implies that it attains its maximum at the left boundary $x = \delta_o$.

Hence, for all $a \in (0,1]$,

$$\begin{aligned}\max_{0\le b\le 1} T(a,b) &\le \max_{0<bp+(1-p)a<1} T(a,b) \\ &\begin{cases} < -(1-p)H(a) & \text{, or} \\ \le (1-R_o)[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)] \end{cases}\end{aligned} \tag{A.4}$$

From (2.24) and the assumption that $R_o < S_2$, we see that

$$(1-R_o)[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)] < 0. \tag{A.5}$$

Also, we have

$$\begin{aligned}(1-R_o)[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)] &\ge (1-R_o)(\beta-1)\ln 2 && \text{(A.6a)} \\ &> -(1-p)\ln 2. && \text{(A.6b)}\end{aligned}$$

Now, since $-(1-p)H(a)$ is a convex function taking values in $[-(1-p)\ln 2, 0]$ and is symmetric about $a = 1/2$, we can conclude from (A.5) and (A.6) that there exists a $\delta_p \in (0, 1/2)$, such that

$$-(1-p)H(a) \le (1-R_o)[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)] \tag{A.7}$$

if and only if $a \in [\delta_p, 1-\delta_p]$. Equations (2.13), (A.4) and (A.7) then prove this part of the theorem after some simple algebraic manipulations.

3. From (2.11),

$$\max_{0<l<n_p\delta_p} \overline{N_p^{ub}(l)} \le \max_{0<l<n_p\delta_p} \max_{0\le i\le np} n\frac{\binom{pn}{i}\binom{(1-p)n}{l}}{\binom{n}{i+l}}\overline{N_o(i+l)} \tag{A.8}$$

Let

$$f(i+l) \triangleq n\frac{\binom{pn}{i}\binom{(1-p)n}{l}}{\binom{n}{i+l}}\overline{N_o(i+l)}. \tag{A.9}$$

We study the behavior of $f(i+l)$ in two cases. When $(i+l) \in (0, n\delta_o)$, we have from Fact 2.1

$$f(i+l) = n\frac{\binom{pn}{i}\binom{(1-p)n}{l}}{\binom{n}{i+l}}O(n^{-j+2}) = O(n_p^{-j+3}), \tag{A.10}$$

where we have used the fact that $\frac{\binom{pn}{i}\binom{(1-p)n}{l}}{\binom{n}{i+l}} \leq 1$ and that $n_p = (1-p)n$. On the other hand, when $(i+l) \in [n\delta_o, n_p\delta_p + np)$, we have from (2.13), (A.3), (A.7), and the fact that $l \in (0, n_p\delta_p)$

$$\lim_{n_p\to\infty} \frac{1}{n_p} \ln f(i+l) \leq H(a) + \frac{(1-R_o)[(\beta-1)\ln 2 + \ln(1+(1-2\delta_o)^k)]}{1-p} \tag{A.11a}$$

$$< H(a) + \frac{-(1-p)H(a)}{1-p} \tag{A.11b}$$

$$= 0 \tag{A.11c}$$

i.e., $f(i+l)$ decreases exponentially in n_p in this range. Hence, from (A.10) and (A.11), this part of the theorem is proved.

4. This part follows from (2.11), Fact 2.1, and the fact that

$$\binom{n}{i} = \binom{n}{n-i}, \forall i \in \{0, 1, \ldots, n\}. \tag{A.12}$$

A.2 Proof of Theorem 2.3

Based on Theorem 2.2, for all $R_o \in (0, S_2)$ the code with rate R_o can be punctured to a code with rate $R_p \in [R_o, R_1]$ and have all four properties listed in Theorem 2.2. We seek sufficient conditions for ϵ such that the punctured code has a vanishing probability of block error with ML decoding on the MBIOS channel with capacity $C = \frac{R_p}{1-\epsilon}$.

Let $U \subset \{1, 2, \ldots, n\}$, and U^c be its complementary set. The following upper bound on the average block error probability under ML decoding was derived in [48]:

$$P_B \leq \sum_{l \in U} \{\overline{N_p(l)} D^l\} + 2^{-n_p E_r(R_p + \frac{\ln \alpha}{n_p \ln 2})}, \tag{A.13}$$

where

$$\alpha \triangleq \max_{l \in U^c} \frac{\overline{N_p(l)}}{2^{n_p R_p} - 1} \frac{2^{n_p}}{\binom{n_p}{l}}, \tag{A.14}$$

$E_r(\cdot)$ is the random coding exponent, and

$$D \triangleq \int_{-\infty}^{\infty} \sqrt{p(y|0)p(y|1)} dy \tag{A.15}$$

is the Bhattacharya parameter, where $p(y|0)$ and $p(y|1)$ are the conditional probability density functions of the output of the MBIOS channel given the input. Note that $D < 1$ since $C > 0$. If we apply this bound to the punctured ensemble and let

$$U \triangleq \left\{ l : \frac{l}{n_p} \in (0, \delta_p) \cup (1 - \delta_p, 1] \right\}, \tag{A.16}$$

then we have from Theorem 2.2 and the fact that $D < 1$,

$$\sum_{l\in U}\{\overline{N_p(l)}D^l\} \leq D^{n_p} + \sum_{l\in U\setminus\{n_p\}} \overline{N_p^{ub}(l)}D^l \leq n_p O\left(n_p^{-j+3}\right) = O\left(n_p^{-j+4}\right). \quad \text{(A.17)}$$

Note that, although there is always one codeword with weight $n_p \in U$, the pairwise error probability due to that codeword in the union bound decreases at least exponentially in n_p.

¿From Theorem 2.2 we have

$$\begin{aligned}
\lim_{n_p\to\infty} \frac{\ln \alpha}{n_p} &= \max_{\delta_p\leq a\leq 1-\delta_p} w_p(a) - [H(a) - (1-R_p)\ln 2] \\
&\leq \max_{\delta_p\leq a\leq 1-\delta_p} w_p^{ub}(a) - [H(a) - (1-R_p)\ln 2] \\
&\leq \frac{(1-R_o)R_p}{R_o}\ln[1+(1-2\delta_o)^k] \\
&\leq \frac{(1-R_o)C}{R_o}\ln[1+(1-2\delta_o)^k]. \quad \text{(A.18)}
\end{aligned}$$

Hence, by (A.13), (A.17) and (A.18), for a punctured ensemble with rate $R_p = (1-\epsilon)C$ to have a vanishing average block error probability, it is sufficient to have

$$(1-\epsilon)C + \frac{\ln\alpha}{n_p\ln 2} < C \Leftrightarrow \epsilon > \frac{\ln\alpha}{n_p C\ln 2} \quad \Leftarrow \epsilon > \frac{(1-R_o)}{R_o\ln 2}\ln[1+(1-2\delta_o)^k]. \quad \text{(A.19)}$$

We would like to further upper bound the quantity on the right hand side of the above inequality, in order to express the sufficient condition only as a function of R_o and not δ_o. Towards this goal, we have from Lemma 2.4 and Lemma 2.2 that for any

$\eta > 1$, there exists an S_1, such that for all $R_o \in (0, S_1)$,

$$\begin{aligned}
\frac{(1-R_o)}{R_o \ln 2} \ln[1 + (1-2\delta_o)^k] \leq & \frac{1}{R_o \ln 2}(1-2\delta_o)^k && \text{(A.20a)} \\
\leq & \frac{1}{R_o \ln 2}\left(1 - \frac{H(\delta_o)}{\ln 2}\right)^{k/3} && \text{(A.20b)} \\
\leq & \frac{1}{R_o \ln 2}(\eta R_o)^{k/3} && \text{(A.20c)} \\
= & \frac{\eta^{k/3} R_o^{k/3-1}}{\ln 2}. && \text{(A.20d)}
\end{aligned}$$

Now, Since $S_2 \leq S_1$ as shown in the proof of Theorem 2.1, this theorem is proved.

APPENDIX B

Proofs of Properties of LDPC-GM Codes in Chapter 3

B.1 Proof of Theorem 3.3

Fix an R_1 and pick an arbitrary $\delta \in (0, H^{-1}((1-R_1)\ln 2))$.

1. Define

$$f(b) \triangleq w_o(b) + a \ln \frac{1-(1-2b)^k}{2} + (1-a)\ln \frac{1+(1-2b)^k}{2}. \tag{B.1}$$

We will bound $f(b)$ in two cases.

Case (a): Let

$$M_1' \triangleq \frac{\ln\left[1-\frac{H(\delta)}{(1-R_1)\ln 2}\right]}{\ln(1-2\delta)} \geq \frac{\ln\left[1-\frac{H(\delta)}{(1-R)\ln 2}\right]}{\ln(1-2\delta)}, \tag{B.2}$$

and $M_1 = \max\{M_1', 1/\delta\}$. By Lemma 2.3, if $k > M_1$ then $w_o(\delta) < 0$ for all

$R \in [0, R_1]$. Therefore, for $k > M_1$ and $b \in [a/k, \delta] \cup [1-\delta, 1-a/k]$, we have

$$f(b) \le w_o(b) - H(a) \le \max\{w_o(a/k), w_o(\delta)\} - H(a), \tag{B.3}$$

where the first inequality follows from the fact that relative entropy is always nonnegative [49, Theorem 2.6.3], and the second inequality follows from Fact 2.1.

Case (b): $b \in (\delta, 1-\delta)$. We have from Lemma 2.1 that

$$\begin{aligned} f(b) \le & (1-R)\ln[1+(1-2b)^k] + H(b) - (1-R)\ln 2 + \\ & + a\ln\frac{1-(1-2b)^k}{2} + (1-a)\ln\frac{1+(1-2b)^k}{2} && \text{(B.4a)} \\ \le & -(1-R)\ln 2 - \ln 2 + \left\{H(b) + (2-R-a)\ln[1+(1-2b)^k]\right\} && \text{(B.4b)} \\ \le & -(1-R)\ln 2 - \ln 2 + \{H(b) + 2\ln[1+(1-2b)^k]\}, && \text{(B.4c)} \end{aligned}$$

where the last two inequalities follow from (3.31). Since

$$\frac{\partial^2 H(b)}{\partial b^2} = -\frac{1}{(1-b)b} \le -4, \tag{B.5}$$

and

$$\begin{aligned} \frac{\partial^2 2\ln[1+(1-2b)^k]}{\partial b^2} = & \frac{8k[k-1-(1-2b)^k](1-2b)^{k-2}}{[1+(1-2b)^k]^2} && \text{(B.6a)} \\ \le & 8k(k-1)(1-2b)^{k-2} && \text{(B.6b)} \\ \le & 8k(k-1)(1-2\delta)^{k-2}, && \text{(B.6c)} \end{aligned}$$

which can be made arbitrarily close to 0 for a large enough k, there exists an

M_2 such that

$$k > M_2 \Rightarrow \frac{\partial^2 H(b) + 2\ln[1+(1-2b)^k]}{\partial b^2} < 0, \quad \forall b \in (\delta, 1-\delta). \tag{B.7}$$

Furthermore, since

$$\left.\frac{\partial H(b) + 2\ln[1+(1-2b)^k]}{\partial b}\right|_{b=1/2} = \left.\left\{\ln\frac{1-b}{b} + \frac{-4k(1-2b)^{k-1}}{1+(1-2b)^k}\right\}\right|_{b=1/2} = 0, \tag{B.8}$$

it follows that the maximum of $H(b)+2\ln[1+(1-2b)^k]$ is attained at $b=1/2$, and thus

$$f(b) \leq -(1-R)\ln 2, \quad \forall b \in (\delta, 1-\delta). \tag{B.9}$$

Based on the above two cases, we have shown that for a fixed R_1 and an arbitrary $\delta \in (0, H^{-1}((1-R_1)\ln 2))$, there exists an $M \triangleq \max\{M_1, M_2\}$ such that for all $k > M$ and for all LDPC-GM ensembles with $R \leq R_1$ we have

$$w_c^{ub}(a) = H(a) + \max_{\frac{a}{k} \leq b \leq 1-\frac{a}{k}} f(b) \leq \max\{H(a) - (1-R)\ln 2, w_o(a/k), w_o(\delta)\}. \tag{B.10}$$

For $a = 0$, we have $\max\{w_o(a/k), w_o(\delta)\} = 0$ and $H(a) - (1-R)\ln 2 = -(1-R)\ln 2 < 0$, which implies $w_c^{ub}(0) \leq 0$. For $a = 1/2$ we have $\max\{w_o(a/k), w_o(\delta)\} < 0$ and $H(a) - (1-R)\ln 2 = R\ln 2 > 0$. Since $\max\{w_o(a/k), w_o(\delta)\} < 0$ for all $a > 0$, $H(a) - (1-R)\ln 2$ must intersect with $\max\{w_o(a/k), w_o(\delta)\}$ at some $a = \delta' < H^{-1}((1-R)\ln 2)$. This is true since $\max\{w_o(a/k), w_o(\delta)\}$ is a monotonically decreasing function with respect to a for all $a \in [0, 1/2]$.

Thus for all $a \in (0, \delta']$, $w_c^{ub}(a) = \max\{w_o(a/k), w_o(\delta)\} < 0$. In addition, for all $a \in [\delta', 1/2]$, $w_c^{ub}(a) = H(a) - (1-R)\ln 2$. This concludes the proof of the first statement of the theorem.

2. For all $l \in (0, \delta' n] \cup [n - \delta' n, n]$ and $k > M$, we have

$$\overline{N_c^{ub}(l)} = \sum_{s=\lceil l/k \rceil}^{\lfloor n-l/k \rfloor} \frac{\overline{N_o(s)}\overline{Z_{s,l}^{(LDGM)}}}{\binom{n}{s}} \tag{B.11a}$$

$$\overset{(a)}{\leq} \sum_{s=\lceil l/k \rceil}^{\delta' n} \overline{N_o(s)} + \sum_{s=n-\delta' n}^{\lfloor n-l/k \rfloor} \overline{N_o(s)} + \sum_{s=\delta' n}^{n-\delta' n} \frac{\overline{N_o(s)}\overline{Z_{s,l}^{(LDGM)}}}{\binom{n}{s}} \tag{B.11b}$$

$$\overset{(b)}{\leq} O(n^{-j+2}) + n\exp\{n[H(l/n) + \max_{\delta' \leq b \leq 1-\delta'} f(b)] + o(n)\} \tag{B.11c}$$

$$\overset{(c)}{\leq} O(n^{-j+2}) + n\exp\{n[H(l/n) - (1-R)\ln 2] + o(n)\} \tag{B.11d}$$

$$\overset{(d)}{=} O(n^{-j+2}), \tag{B.11e}$$

where $o(n)$ denotes some value that converges to 0 as n approaches infinity. In (B.11), (a) follows from the fact that $\overline{Z_{s,l}^{(LDGM)}}/\binom{n}{s} \leq 1$ since it is a probability as shown in (3.4); (b) follows from Fact 2.1; (c) follows from (B.9); (d) follows from the fact that $\delta' < H^{-1}((1-R)\ln 2)$.

B.2 Proof of Theorem 3.4

Let M be as defined in Theorem 3.3 for $R_1 = C$. Moreover, let $U \subset \{1, 2, \ldots, n\}$, and U^c be its complementary set. The following upper bound on the average block error probability under ML decoding is given in [48]:

$$P_B \leq \sum_{l \in U} \{\overline{N_c(l)} D^l\} + 2^{-nE_r(R + \frac{\ln \alpha}{n \ln 2})}, \tag{B.12}$$

where

$$\alpha \triangleq \max_{l \in U^c} \frac{\overline{N_c(l)}}{2^{nR}-1} \frac{2^n}{\binom{n}{l}}, \tag{B.13}$$

$E_r(\cdot)$ is the random coding exponent, and

$$D \triangleq \int_{-\infty}^{\infty} \sqrt{p(y|0)p(y|1)}dy \leq 1 \tag{B.14}$$

is the Bhattacharyya parameter, where $p(y|0)$ and $p(y|1)$ are the conditional probability density functions of the output of the MBIOS channel given the input. We will apply this bound to the LDPC-GM ensemble with $k > M$, $R < C$, and

$$U \triangleq \left\{ l : \frac{l}{n} \in (0, \delta'] \cup [1-\delta', 1] \right\}, \tag{B.15}$$

where δ' is as defined in Theorem 3.3. Regarding the first term, we have from Theorem 3.3 that

$$\sum_{l \in U} \{\overline{N_c(l)} D^l\} \leq \sum_{l \in U} \overline{N_c^{ub}(l)} \leq nO(n^{-j+2}) = O(n^{-j+3}). \tag{B.16}$$

Regarding the second term, we have from the same theorem and Lemma 3.1 that

$$\lim_{n\to\infty} \frac{\ln a}{n} = \max_{a \in (\delta', 1-\delta')} w_c(a) - [H(a) - (1-R)\ln 2] \tag{B.17a}$$

$$\leq \max_{a \in (\delta', 1/2]} w_c^{ub}(a) - [H(a) - (1-R)\ln 2] \tag{B.17b}$$

$$\leq 0. \tag{B.17c}$$

Hence

$$P_B \leq O(n^{-j+3}) + 2^{-nE_r(R)}, \tag{B.18}$$

which converges to 0 as n approaches infinity for all $R < C$ and $j \geq 4$. Thus, the theorem is proved.

B.3 Proofs of Section 3.4

First, we need a lemma.

Lemma B.1 *If the degree distribution pair* (λ, ρ) *satisfies* $\rho(0) = 0$, $\rho(1) = 1$, *and satisfies* (3.41) *for all* $x_3 \in [0, 1]$, *then* $R = 1 - q$.

Proof: [18, Lemma 1] shows that under the assumed conditions, we have

$$\frac{\int_0^1 \rho(t)dt}{\int_0^1 \lambda(t)dt} = q. \tag{B.19}$$

■

B.3.1 Proof of Theorem 3.5

The facts that (λ, ρ) satisfies (3.41) for all $x \in [0, 1]$ and that $\lambda(x)$ has only non-negative coefficients for $k = 3$ and $q \in [\frac{12}{13}, 1)$ are proved in [18, Theorem 1]. By the definition of λ_ϵ, we have effectively

$$\lambda_\epsilon(x) = \sum_{i=1}^{M(\epsilon)} \lambda_i x^{i-1} \tag{B.20}$$

in the density evolution equations. Hence, it follows that $\lambda_\epsilon(x) < \lambda(x)$, and the corresponding $\tilde{\lambda}_\epsilon(x) < \tilde{\lambda}(x)$ for all $x \in (0, 1]$. Therefore, (3.43) is satisfied, which implies that the BP decoding is successful. To find the rate of this ensemble of codes, let

$$\delta \triangleq \sum_{M(\epsilon)+1}^{\infty} \tilde{\lambda}_i \tag{B.21}$$

be the fraction of pilot nodes. Then, we have

$$R = \frac{(1-\delta)\int_0^1 \lambda(t)dt - \int_0^1 \rho(t)dt}{\int_0^1 \lambda(t)dt} \tag{B.22a}$$

$$= 1 - \delta - \frac{\int_0^1 \rho(t)dt}{\int_0^1 \lambda(t)dt} \tag{B.22b}$$

$$= 1 - q - \delta, \tag{B.22c}$$

where the last equality follows from the facts that $\rho(0) = 0$, $\rho(1) = 1$, and Lemma B.1. But, from (3.39)

$$\delta = \sum_{M(\epsilon)+1}^{\infty} \frac{\lambda_i/i}{\int_0^1 \lambda(t)dt} = q \sum_{M(\epsilon)+1}^{\infty} \frac{\lambda_i/i}{\int_0^1 \rho(t)dt} = qk \sum_{M(\epsilon)+1}^{\infty} \lambda_i/i < \epsilon(1-q). \tag{B.23}$$

Therefore, it follows that $R > (1-\epsilon)(1-q)$, and the theorem is proved.

B.3.2 Proof of Theorem 3.6

The facts that (λ, ρ) satisfies (3.41) for all $x \in [0, 1]$ and that $\rho(x)$ has only non-negative coefficients for $q \in [0.05, 1]$ are proved in [18, Theorem 2]. Since $\rho_\epsilon(x) > \rho(x)$ for all $x \in (0, 1]$, (3.43) is satisfied and the BP decoding is successful. As for the

rate of this ensemble of codes, we have

$$
\begin{aligned}
R =& 1 - \frac{\int_0^1 \rho_\epsilon(t)dt}{\int_0^1 \lambda(t)dt} && \text{(B.24a)}\\
=& 1 - \frac{\sum_{i=1}^{M(\epsilon)} \frac{\rho_i}{i} + 1 - \sum_{i=1}^{M(\epsilon)} \rho_i}{\int_0^1 \lambda(t)dt} && \text{(B.24b)}\\
>& 1 - \frac{\sum_{i=1}^{\infty} \frac{\rho_i}{i} + 1 - \sum_{i=1}^{M(\epsilon)} \rho_i}{\int_0^1 \lambda(t)dt} && \text{(B.24c)}\\
=& 1 - \frac{\int_0^1 \rho(t)dt + \sum_{i=M(\epsilon)+1}^{\infty} \rho_i}{\int_0^1 \lambda(t)dt} && \text{(B.24d)}\\
\overset{(a)}{=}& 1 - q - 3 \sum_{i=M(\epsilon)+1}^{\infty} \rho_i && \text{(B.24e)}\\
>& (1-\epsilon)(1-q), && \text{(B.24f)}
\end{aligned}
$$

where (a) follows from the facts that $\rho(0) = 0$, $\rho(1) = 1$, and Lemma B.1. Hence, the theorem is proved.

B.3.3 Proof of Theorem 3.7

Let F be the degree distribution from the node perspective[1] corresponding to f. We have

$$F(x) = \frac{\int_0^x f(t)dt}{\int_0^1 f(t)dt} = [x(1-p)+p]^2, \tag{B.25}$$

[1]That is, $F(x) = \sum_{i=0}^{\infty} F_i x^i$,where F_i denotes the fraction of input nodes that have i neighboring check nodes in the LDGM code.

and the following set of density evolution equations

$$x_1 = 1 - (1-q)(1-x_4) \tag{B.26a}$$

$$x_2 = F(x_1)\lambda(x_3) \tag{B.26b}$$

$$x_3 = 1 - \rho(1-x_2) \tag{B.26c}$$

$$x_4 = f(x_1)\tilde{\lambda}(x_3), \tag{B.26d}$$

where q denotes the channel erasure probability. After some algebraic manipulations, the fixed point equation can be shown to be

$$x_3 = 1 - \rho\left(1 - \left[\frac{q(1-p)+p}{1-(1-q)(1-p)\tilde{\lambda}(x_3)}\right]^2 \lambda(x_3)\right), \tag{B.27}$$

which is the same as (3.41) if we let the erasure probability be $q' = q(1-p)+p$. Hence, from Theorem 3.5 and Theorem 3.6, the decoding is successful under BP decoding on the BEC with erasure probability q. Moreover, the rate of this ensemble is given by

$$R = \{\text{rate of the outer LDPC code}\} \times \frac{\{\text{number of input nodes in the LDGM code}\}}{\{\text{number of check nodes in the LDGM code}\}} \tag{B.28a}$$

$$= \{\text{rate of the outer LDPC code}\} \times \frac{G'(1)}{F'(1)} \tag{B.28b}$$

$$> (1-\epsilon)(1-q')\frac{1}{1-p} \tag{B.28c}$$

$$= (1-\epsilon)(1-q), \tag{B.28d}$$

which then proves this theorem.

APPENDIX C

Proof of Analytical Results for Algorithms in Chapter 5

C.1 Constant Phase Model: Exact GLRT Algorithm

Let $\mathbf{s}_0$, $\hat{\mathbf{s}}_{CSI}$ be the transmitted sequence and the detected sequence for the hypothetical receiver that has perfect CSI, respectively. The probability of sequence error for the algorithm can be bounded as

$$
\begin{aligned}
&P_{MLSD} \\
=&P(\hat{\mathbf{s}}_{MLSD} \neq \mathbf{s}_0) && \text{(C.1a)} \\
=&P(\hat{\mathbf{s}}_{MLSD} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{CSI} = \mathbf{s}_0) + P(\hat{\mathbf{s}}_{MLSD} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{CSI} \neq \mathbf{s}_0) && \text{(C.1b)} \\
\leq&P(\{\hat{\mathbf{s}}_{MLSD} \neq \mathbf{s}_0\} \cap \{\mathbf{s}_0 = \mathbf{s}_j \text{ for some } 1 \leq j \leq 2(N-1)\}) + P(\hat{\mathbf{s}}_{CSI} \neq \mathbf{s}_0) && \text{(C.1c)} \\
\leq&P\left(\bigcup_{\substack{1\leq i\leq 2(N-1),\\ i\neq j}} \{|\mathbf{z}^H\mathbf{D}\mathbf{s}_0| \leq |\mathbf{z}^H\mathbf{D}\mathbf{s}_i|\}\right) + 1 - \left[1 - Q\left(\sqrt{\frac{2E_s}{N_0}}\right)\right]^{N-1}, && \text{(C.1d)}
\end{aligned}
$$

where $\mathbf{s}_i, \forall i \neq 0$ are the candidate sequences as defined in the exact GLRT algorithm. To further evaluate the first term of (C.1), we define

$$X \triangleq \mathbf{z}^H \mathbf{D}\mathbf{s}_0, \tag{C.2}$$

$$Y_i \triangleq \mathbf{z}^H \mathbf{D}\mathbf{s}_i \quad \forall 1 \leq i \leq 2(N-1). \tag{C.3}$$

Since X and $Y_i's$ are jointly Gaussian, we have the conditional pdf of Y_i given X as

$$f_{Y_i|X}(y|X) = \mathcal{CN}\left(y, \frac{E_t - 2w_i E_s}{E_t} X, \frac{4w_i E_s (E_t - w_i E_s) N_0}{E_t}\right), \tag{C.4}$$

where $\mathcal{CN}(\cdot, m, \sigma^2)$ is the pdf of a circularly symmetric complex Gaussian random variable with mean m and variance σ^2, and w_i is the number of places where $\mathbf{s}_0$ and $\mathbf{s}_i$ differ. Therefore, we have

$$\begin{aligned}
&P\left(\left|\mathbf{z}^H \mathbf{D}\mathbf{s}_0\right| \leq \left|\mathbf{z}^H \mathbf{D}\mathbf{s}_i\right|\right) \\
&= \int_{\mathcal{C}} P(|Y_i| \geq |x| | X = x) f_X(x) dx && \text{(C.5a)} \\
&= \int_{\mathcal{C}} \int_{|x|}^{\infty} R\left(r, \left|\frac{E_t - 2w_i E_s}{E_t}\right| |x|, \frac{2w_i E_s (E_t - w_i E_s) N_0}{E_t}\right) f_X(x) dr dx && \text{(C.5b)} \\
&= \int_0^{\infty} R\left(r, E_t, \frac{E_t N_0}{2}\right) Q_1\left(\frac{r|E_t - 2w_i E_s|}{\sqrt{2w_i E_t E_s (E_t - w_i E_s) N_0}}, \frac{r\sqrt{E_t}}{\sqrt{2w_i E_s (E_t - w_i E_s) N_0}}\right) dr, && \text{(C.5c)}
\end{aligned}$$

where $\mathcal{C}$ is the set of all complex numbers and f_X is the pdf of X. Hence, by taking the union bound in (C.1) and use the fact that candidate sequences corresponding to neighboring partitions of $[0, 2\pi)$ differ in only one symbol, which follows from the structure of the exact polynomial-complexity algorithm, we obtain (5.23).

C.2 Constant Phase Model: Pilot-Only (PO) Algorithm

Since

$$\begin{aligned}
&\arg\max_{\tilde{s}\in\mathcal{A}} p(z_i|\tilde{s}, z_1) \\
=&\arg\max_{\tilde{s}\in\mathcal{A}} \int_0^{2\pi} p(z_i|\tilde{s}, \theta, z_1) f(\theta|z_1) d\theta && \text{(C.6a)} \\
=&\arg\max_{\tilde{s}\in\mathcal{A}} \int_0^{2\pi} e^{\frac{-|z_i-\tilde{s}\sqrt{E_s}e^{j\theta}|^2}{N_0}} \frac{f(z_1|\theta)f(\theta)}{f(z_1)} d\theta && \text{(C.6b)} \\
=&\arg\max_{\tilde{s}\in\mathcal{A}} \int_0^{2\pi} e^{\frac{2\Re\{z_i\tilde{s}\sqrt{E_s}e^{-j\theta}\}}{N_0}} e^{\frac{2|z_1|\sqrt{E_p}\cos(\angle z_1-\theta)}{N_0}} d\theta \quad \forall i = 2, 3, \ldots, N, && \text{(C.6c)}
\end{aligned}$$

where f denotes the pdf's dictated by its parameters, the bit error probability of the algorithm can be evaluated as follows

$$\begin{aligned}
&P_b(PO) \\
=&P\left(\int_0^{2\pi} e^{\frac{2\Re\{z\sqrt{E_s}e^{-j\theta}\}}{N_0}} e^{\frac{2|z_1|\sqrt{E_p}\cos(\angle z_1-\theta)}{N_0}} d\theta < \int_0^{2\pi} e^{\frac{-2\Re\{z\sqrt{E_s}e^{-j\theta}\}}{N_0}} e^{\frac{2|z_1|\sqrt{E_p}\cos(\angle z_1-\theta)}{N_0}} d\theta\right) && \text{(C.7a)} \\
=&P\left(\int_0^{2\pi} \left[e^{\frac{2r\sqrt{E_s}\cos(\theta-\angle z)}{N_0}} - e^{\frac{-2r\sqrt{E_s}\cos(\theta-\angle z)}{N_0}}\right] e^{\frac{2|z_1|\sqrt{E_p}\cos(\angle z_1-\theta)}{N_0}} d\theta < 0\right) && \text{(C.7b)} \\
=&P\left(\int_0^{2\pi} \sinh\left(\frac{2r\sqrt{E_s}\cos(t-\theta')}{N_0}\right) e^{\frac{2|z_1|\sqrt{E_p}\cos\theta'}{N_0}} d\theta' < 0\right), && \text{(C.7c)}
\end{aligned}$$

where z is a circularly symmetric complex Gaussian random variable with mean $\sqrt{E_s}$ and variance N_0, $r \triangleq |z|$, and $t \triangleq \angle z_1 - \angle z$. Since $\sinh(\frac{2r\sqrt{E_s}\cos\theta}{N_0})$ and $e^{\frac{2|z_1|\sqrt{E_p}\cos\theta}{N_0}}$ are both symmetric bell shaped functions of θ in one period $[-\pi, \pi]$, their circular convolution would still be a bell shaped function. Observing further that $\int_0^{2\pi} \sinh(\frac{2r\sqrt{E_s}\cos(t-\theta')}{N_0}) e^{\frac{2|z_1|\sqrt{E_p}\cos\theta'}{N_0}} d\theta'$ attains maximum at $t = 0$, minimum at

$t = \pi$, and 0 at $t = \pm\frac{\pi}{2}$, we have

$$P_b(PO) = 1 - P\left(t \in [-\frac{\pi}{2}, \frac{\pi}{2}]\right) \tag{C.8a}$$

$$= 1 - E_\theta\left[\int_0^{2\pi} p\left(\angle z \in [-\frac{\pi}{2} + \angle z_1, \frac{\pi}{2} + \angle z_1] | \angle z_1, \theta\right) f(\angle z_1|\theta) d\angle z_1\right] \tag{C.8b}$$

$$= 1 - E_\theta\left[\int_0^{2\pi} \int_{-\frac{\pi}{2}+x}^{\frac{\pi}{2}+x} T\left(y - \theta, \frac{E_s}{N_0}\right) dy T\left(x - \theta, \frac{E_p}{N_0}\right) dx\right] \tag{C.8c}$$

$$= 1 - E_\theta\left[\int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T\left(y' + x', \frac{E_s}{N_0}\right) T\left(x', \frac{E_p}{N_0}\right) dy' dx'\right] \tag{C.8d}$$

$$= 1 - \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} T\left(y' + x', \frac{E_s}{N_0}\right) T\left(x', \frac{E_p}{N_0}\right) dy' dx'. \tag{C.8e}$$

It then follows that

$$P_{PO} = 1 - \left[1 - P_b(PO)\right]^{N-1}. \tag{C.9}$$

C.3 Constant Phase Model: Uniform Sampling (US) Algorithm

The lower bound follows easily by the optimality of the MLSD metric. The upper bound of P_{US} is derived as follows

$$P_{US} = P(\hat{\mathbf{s}}_{US} \neq \mathbf{s}_0) \tag{C.10a}$$

$$= P(\hat{\mathbf{s}}_{US} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{MLSD} = \mathbf{s}_0) + P(\hat{\mathbf{s}}_{US} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{MLSD} \neq \mathbf{s}_0) \tag{C.10b}$$

$$\leq P(\hat{\mathbf{s}}_{US} \neq \hat{\mathbf{s}}_{MLSD} = \mathbf{s}_0) + P_{MLSD}, \tag{C.10c}$$

where $\mathbf{s}_0$ is the transmitted sequence. The first term of this upper bound can be further bounded as

$$
\begin{aligned}
&P(\hat{\mathbf{s}}_{US} \neq \hat{\mathbf{s}}_{MLSD} = \mathbf{s}_0) \\
\leq &P\left(\text{No sample point is in the same partition with } \hat{\theta}(\mathbf{s}_0)\right) && \text{(C.11a)} \\
= &P\left(\bigcup_{i=0}^{L-1} \bigcup_{k,l:k\neq l} \{\phi_k, \phi_l \in (a_i, a_{i+1})\} \cap \{\hat{\theta}(\mathbf{s}_0) \in (\phi_k, \phi_l)\}\right) && \text{(C.11b)} \\
\leq &\sum_{i=1}^{L} \sum_{k,l:k\neq l} E_\theta\left[P\left(\{\phi_k, \phi_l \in [a_i, a_{i+1}]\} \cap \{\hat{\theta}(\mathbf{s}_0) \in (\phi_k, \phi_l)\}|\theta\right)\right], && \text{(C.11c)}
\end{aligned}
$$

where the notation $\phi_l \in [a, b]$ is used to denote the event that one of the threshold pairs obtained from z_l as in (5.8) is inside $[a, b]$. Define

$$
\begin{aligned}
x_1 \triangleq &E_p e^{j\theta} + \sqrt{E_p} n_1, && \text{(C.12)} \\
x_i \triangleq &E_s e^{j\theta} + s_{0,i} n_i \quad \forall i = 2, 3, \ldots, N, && \text{(C.13)} \\
X \triangleq &\sum_{i=1}^{N} x_i. && \text{(C.14)}
\end{aligned}
$$

We have

$$
\begin{aligned}
\phi_i = &\angle z_i \pm \frac{\pi}{2} = s_{0,i} \angle z_i \pm \frac{\pi}{2} = \angle x_i \pm \frac{\pi}{2} \quad \forall i = 2, 3, \ldots, N, && \text{(C.15)} \\
\hat{\theta}(\mathbf{s}_0) = &-\angle(\mathbf{z}^H \mathbf{s}_0) = \angle \sum_{i=1}^{N} \left(s_{0,i}^2 e^{j\theta} + s_{0,i} n_i\right) = \angle X. && \text{(C.16)}
\end{aligned}
$$

Hence

$$
\begin{aligned}
&p(\{\phi_k, \phi_l \in [a_i, a_{i+1}]\} \cap \{\hat{\theta}(\mathbf{s}_0) \text{ is between } \phi_k, \phi_l\}|\theta) \\
= &2 \int_{X:\angle X \in [a_i, a_{i+1}]} \int_{x_k:\angle x_k \pm \frac{\pi}{2} \in [a_i, \angle X]} \int_{x_l:\angle x_l \pm \frac{\pi}{2} \in [\angle X, a_{i+1}]} f(x_k, x_l, X|\theta) dx_l dx_k dX, && \text{(C.17)}
\end{aligned}
$$

which is under the assumption that $L \geq 2$ so that the ith pair of thresholds can not lie in the same sampling interval for all i. Since x_i's are iid complex Gaussian random variables and X is the sum of them, we have

$$f(X|\theta) = |X| f(|X|e^{j\angle X}|\theta) = \frac{|X|}{\pi E_t N_0} e^{-\frac{||X|e^{j\angle X} - E_t e^{j\theta}|^2}{E_t N_0}}, \tag{C.18}$$

$$f(x_l|X,\theta) = \mathcal{CN}(x_l, \frac{E_s}{E_t}X, E_s N_0(1-\frac{E_s}{E_t}), \tag{C.19}$$

$$f(x_k|x_l,X,\theta) = \mathcal{CN}(x_k, \frac{E_s}{E_p+(N-2)E_s}(X-x_l), E_s N_0(1-\frac{E_s}{E_p+(N-2)E_s}). \tag{C.20}$$

Letting

$$E_{N-i} \triangleq \frac{[\frac{E_s}{E_p+(N-i)E_s}]^2}{E_s(1-\frac{E_s}{E_p+(N-i)E_s})} \quad \text{for } i \text{ integer}, \tag{C.21}$$

the expectation in the above equation can be exactly evaluated as

$$2\int_0^\infty \int_{a_i}^{a_{i+1}} \left[\int_{x_l:\angle x_l \pm \frac{\pi}{2} \in [\angle X, a_{i+1}]} \int_{\theta_k \pm \frac{\pi}{2} \in [a_i, \angle X]} T(\theta_k - \angle(X-x_l), \frac{E_{N-2}}{N_0}|X|^2) \right.$$
$$\left. \frac{|x_l|}{\pi E_s N_0(1-\frac{E_s}{E_t})} e^{-\frac{||x_l|e^{j\angle x_l} - \frac{E_s}{E_t}X|^2}{E_s N_0(1-\frac{E_s}{E_t})}} d\theta_k dx_l \right] \frac{|X|}{\pi E_t N_0} e^{-\frac{||X|e^{j\angle X} - E_t e^{j\theta}|^2}{E_t N_0}} d\angle X d|X|. \tag{C.22}$$

Now, observe that the integrand of (C.22) depends only on θ, which is uniformly distributed in $[0, 2\pi)$. Since all sampling intervals have the same lengths, (C.22) should not depend on the choice the sampling interval i. Also, note that (C.22) does not depend on k nor l. Hence, after combining all the terms, we obtain

$$P(\hat{\mathbf{s}}_{US} \neq \hat{\mathbf{s}}_{MLSD} = \mathbf{s}_0) \leq L(N-1)(N-2) \int_0^\infty \int_0^{\frac{2\pi}{L}} \frac{1}{2\pi} \left[\int_{-\angle X}^{\angle X} T(\theta_1 - \frac{\pi}{2}, \frac{E_{N-2}}{N_0}|X|^2) d\theta_1 \right]$$
$$\left[\int_{-\frac{2\pi}{L}+\angle X}^{\frac{2\pi}{L}-\angle X} T(\theta_2 - \frac{\pi}{2}, \frac{E_{N-1}}{N_0}|X|^2) d\theta_2 \right] R(|X|, E_t, \frac{E_t N_0}{2}) d\angle X d|X|,$$

which reduces to (5.35a).

C.4 Linear phase Model: 2-Dimensional Uniform Sampling (US2D) Algorithm

The lower bound is obvious. The upper bound can be derived as follows

$$
\begin{aligned}
& P_{US2D} \\
=& P(\hat{\mathbf{s}}_{US2D} \neq \mathbf{s}_0) && \text{(C.23a)} \\
=& P(\hat{\mathbf{s}}_{US2D} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{CSI} = \mathbf{s}_0, \hat{\mathbf{s}}_{GLRT} = \mathbf{s}_0) + P(\hat{\mathbf{s}}_{US2D} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{CSI} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{GLRT} = \mathbf{s}_0) \\
& + P(\hat{\mathbf{s}}_{US2D} \neq \mathbf{s}_0, \hat{\mathbf{s}}_{GLRT} \neq \mathbf{s}_0) && \text{(C.23b)} \\
\leq& P(\hat{\mathbf{s}}_{US2D} \neq \hat{\mathbf{s}}_{GLRT} = \hat{\mathbf{s}}_{CSI}) + P_{CSI} + P_{GLRT}, && \text{(C.23c)}
\end{aligned}
$$

where $\mathbf{s}_0$ is the transmitted sequence. The first term of this upper bound can be further calculated as follows

$$
\begin{aligned}
& P(\hat{\mathbf{s}}_{US2D} \neq \hat{\mathbf{s}}_{GLRT} = \hat{\mathbf{s}}_{CSI}) \\
\leq& P(\text{no sample pair is in the same partition with}(f_d, \theta)) && \text{(C.24a)} \\
=& P(\exists \text{ partitioning lines between } (f_d, \theta) \text{ and the nearest four sample pairs}). \\
& && \text{(C.24b)}
\end{aligned}
$$

To evaluate this expression, we define

$$A \triangleq \{\text{No partitioning line lies between } \left(f_d - \frac{1}{Q_f}, \theta\right) \text{ and } (f_d, \theta)\}, \tag{C.25a}$$

$$B \triangleq \{\text{No partitioning line lies between } (f_d, \theta) \text{ and } \left(f_d, \theta + \frac{2\pi}{Q_\theta}\right)\}, \tag{C.25b}$$

$$C \triangleq \{\text{No partitioning line lies between } (f_d, \theta) \text{and } \left(f_d + \frac{1}{Q_f}, \theta\right)\}, \tag{C.25c}$$

$$D \triangleq \{\text{No partitioning line lies between } \left(f_d, \theta - \frac{2\pi}{Q_\theta}\right) \text{ and } (f_d, \theta)\}. \tag{C.25d}$$

Also, recall that all partitioning lines have negative slopes as in (5.21). We have

$$\begin{aligned} &P(\exists \text{ partitioning lines between } (f_d, \theta) \text{ and the nearest four partitioning points}) \\ \leq &P((A^c \cup B^c) \cap (B^c \cup C^c) \cap (C^c \cup D^c) \cap (D^c \cup A^c)) && \text{(C.26a)} \\ = &P((A^c \cap C^c) \cup (B^c \cap D^c)) && \text{(C.26b)} \\ \leq &P(A^c \cap C^c) + P(B^c \cap D^c). && \text{(C.26c)} \end{aligned}$$

We define further the event

$$\begin{aligned} \mathcal{E}_k(f_1, \theta_1, f_2, \theta_2) \triangleq \{&\exists \text{ partitioning lines corresponding to} \\ &\text{the kth symbol that lie between } (f_1, \theta_1) \text{ and } (f_2, \theta_2)\} \end{aligned} \tag{C.27}$$

and subsequently

$$ql(k) \triangleq P\left(\mathcal{E}_k\left(f_d - \frac{1}{Q_f}, \theta, f_d, \theta\right) \mid f_d, \theta\right), \tag{C.28a}$$

$$qu(k) \triangleq P\left(\mathcal{E}_k\left(f_d, \theta, f_d, \theta + \frac{2\pi}{Q_\theta}\right) \mid f_d, \theta\right), \tag{C.28b}$$

$$qr(k) \triangleq P\left(\mathcal{E}_k\left(f_d, \theta, f_d + \frac{1}{Q_f}, \theta\right) \mid f_d, \theta\right), \tag{C.28c}$$

$$qd(k) \triangleq P\left(\mathcal{E}_k\left(f_d, \theta - \frac{2\pi}{Q_\theta}, f_d, \theta\right) \mid f_d, \theta\right). \tag{C.28d}$$

Since conditioning on f_d and θ the N sets of partitioning lines are independent, we have

$$\begin{aligned}
&P(A^c \cap C^c) + P(B^c \cap D^c) \\
=&[1 - P(A) - P(C) + P(A \cap C)] + [1 - P(B) - P(D) + P(B \cap D)] && \text{(C.29a)} \\
=&E_{f_d,\theta}\left[\left\{1 - \prod_{k=1}^{N}(1 - ql(k)) - \prod_{k=1}^{N}(1 - qr(k)) + \prod_{k=1}^{N}(1 - ql(k) - qr(k))\right\} + \right. \\
&\left. + \left\{1 - \prod_{k=1}^{N}(1 - qu(k)) - \prod_{k=1}^{N}(1 - qd(k)) + \prod_{k=1}^{N}(1 - qu(k) - qd(k))\right\}\right], && \text{(C.29b)}
\end{aligned}$$

where we have used the fact that $Q_\theta \geq 4$ and $Q_f \geq 4N$ in the last equality. Now, since $Q_f \geq 4N$, $ql(k)$ and $qr(k)$ can be evaluated as follows

$$\begin{aligned}
ql(k) =&P\left(\frac{1}{2\pi k}\left(\angle z_k - \theta \pm \frac{\pi}{2}\right) \in \left[f_d - \frac{1}{Q_f}, f_d\right] \mid f_d, \theta\right) && \text{(C.30a)} \\
=&\int_{2\pi f_d k - \frac{2\pi k}{Q_f}}^{2\pi f_d k} T\left(x - 2\pi f_d k - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x - 2\pi f_d k + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx && \text{(C.30b)} \\
=&\int_{0}^{\frac{2\pi k}{Q_f}} T\left(x - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx && \text{(C.30c)} \\
=&qr(k) = O\left(\frac{k}{Q_f} e^{-\frac{E_s}{N_0}}\right). && \text{(C.30d)}
\end{aligned}$$

Similarly, since $Q_\theta \geq 4$, we can evaluate $qu(k)$ and $qd(k)$ as follows

$$qu(k) = P\left(\left(\angle z_k - 2\pi k f_d \pm \frac{\pi}{2}\right) \in \left[\theta, \theta + \frac{2\pi}{Q_\theta}\right] \mid f_d, \theta\right) \tag{C.31a}$$

$$= \int_\theta^{\theta+\frac{2\pi}{Q_\theta}} T\left(x - \theta - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x - \theta + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx \tag{C.31b}$$

$$= \int_0^{\frac{2\pi}{Q_\theta}} T\left(x - \frac{\pi}{2}, \frac{E_s}{N_0}\right) + T\left(x + \frac{\pi}{2}, \frac{E_s}{N_0}\right) dx \tag{C.31c}$$

$$= qd(k) = O\left(\frac{1}{Q_\theta} e^{-\frac{E_s}{N_0}}\right). \tag{C.31d}$$

Note that $ql(k), qr(k), qu(k)$ and $qd(k)$ actually do not depend on f_d and θ, and $qu(k)$ and $qd(k)$ do not depend on k. Consequently, we have

$$\begin{aligned} &P(\hat{\mathbf{s}}_{US2D} \neq \hat{\mathbf{s}}_{GLRT} = \hat{\mathbf{s}}_{CSI}) \\ &\leq \left\{1 - 2\prod_{k=1}^{N}(1 - ql(k)) + \prod_{k=1}^{N}(1 - 2ql(k))\right\} + \left\{1 - 2(1 - qu)^N + (1 - 2qu)^N\right\}, \end{aligned} \tag{C.32}$$

which reduces to the expression in (5.38a).

APPENDIX D

Proofs of Lemmas in Chapter 6

D.1 Proof of Lemma 6.1

Define

$$\hat{i}(c) \triangleq \arg\max_{j\in\mathcal{S}} L\left(\hat{\mathbf{V}}^N(j|c), c\right) \tag{D.1}$$

to be the best survivor given c. We have

$$\hat{\mathbf{V}}^N\left(\hat{i}(c)|c\right) = \arg\max_{\mathbf{e}\in\tilde{\mathcal{E}}^N} L(\mathbf{e}, c). \tag{D.2}$$

Define further

$$\hat{c}(\mathbf{e}) \triangleq \arg\max_{c\in\mathbb{C}} L^N(\mathbf{e}, c). \tag{D.3}$$

Since $\hat{c}(\mathbf{e}) \in \mathbb{C}$, $\forall \mathbf{e} \in \tilde{\mathcal{E}}^N$, (6.5) becomes

$$\hat{\mathbf{e}}_{MAPSqD} = \arg\max_{\mathbf{e}\in\tilde{\mathcal{E}}^N} L^N\left(\mathbf{e}, \hat{c}(\mathbf{e})\right) \tag{D.4a}$$

$$= \hat{\mathbf{V}}^N\left(\hat{i}(c)|c\right) \text{ for some } c \in \mathbb{C}, \tag{D.4b}$$

which proves lemma 6.1.

D.2 Proof of Lemma 6.2

By the VA, we know

$$\begin{aligned}
&\hat{\mathbf{V}}^{k+1}(c) \in ext\left(\hat{\mathbf{V}}^{k}(c)\right) \quad \text{and} \quad \hat{\mathbf{V}}^{k+1}(i,k+1|c) = e \\
\Rightarrow & ns(e) = i \quad \text{and} \quad \hat{\mathbf{V}}^{k+1}(i|c) = [\hat{\mathbf{V}}^{k}(ps(e)|c), e]^T \\
\Rightarrow & L^{k+1}\left([\hat{\mathbf{V}}^{k}(ps(e)|c), e]^T, c\right) = \max_{\tilde{e}\in\mathcal{E}:ns(\tilde{e})=i} L^{k+1}\left([\hat{\mathbf{V}}^{k}(ps(\tilde{e})|c), \tilde{e}]^T, c\right),\ \forall i \in \mathcal{S} \quad \text{(D.5)}
\end{aligned}$$

where $\hat{\mathbf{V}}^{k+1}(i,j|c)$ denotes the jth element of the ith survivor in $\hat{\mathbf{V}}^{k+1}(c)$. Therefore

$$\begin{aligned}
& T^{k+1}(\mathbf{V}^{k+1}) \\
\triangleq & \left\{c \in \mathbb{C} : \hat{\mathbf{V}}^{k+1}(c) = \mathbf{V}^{k+1}\right\} \\
= & \Big\{c \in \mathbb{C} : \mathbf{V}^{k+1} \in ext\left(\mathbf{W}^{k}\right), \mathbf{W}^{k} = \hat{\mathbf{V}}^{k}(c),\ \mathbf{V}^{k+1}(i,k+1) = e \\
& \Rightarrow L^{k+1}\left([\mathbf{W}^{k}(ps(e)), e]^T, c\right) = \max_{\tilde{e}\in\mathcal{E}:ns(\tilde{e})=i} L^{k+1}\left([\mathbf{W}^{k}(ps(\tilde{e})), \tilde{e}]^T, c\right),\ \forall i \in \mathcal{S}\Big\} \\
= & \bigcup_{\mathbf{W}^{k}:\mathbf{V}^{k+1}\in ext(\mathbf{W}^{k})} T^{k}(\mathbf{W}^{k}) \cap \big\{c \in \mathbb{C} : \mathbf{V}^{k+1}(i,k+1) = e \\
& \Rightarrow L^{k+1}\left([\mathbf{W}^{k}(ps(e)), e]^T, c\right) = \max_{\tilde{e}\in\mathcal{E}:ns(\tilde{e})=i} L^{k+1}\left([\mathbf{W}^{k}(ps(\tilde{e})), \tilde{e}]^T, c\right),\ \forall i \in \mathcal{S}\Big\} \\
= & \bigcup_{\mathbf{W}^{k}:\mathbf{V}^{k+1}\in ext(\mathbf{W}^{k})} \left\{T^{k}(\mathbf{W}^{k}) \bigcap_{i\in\mathcal{S}} P^{k}(\mathbf{W}^{k}, \mathbf{V}^{k+1}(i,k+1))\right\}
\end{aligned}$$

which proves lemma 6.2.

D.3 Proof of Lemma 6.3

Assume $I = 2$. Let $i \to j$ denote a transition from state i to state j. As defined in section 6.4, a group at time k is uniquely specified by the following descriptions.

1. The merging time $m < k$ before which all the survivors merge together into one tail.

2. The merging state $s_m \in \mathcal{S} = \{1, 2\}$ to which the only tail connects at the merging time m.

3. The one-to-two transition (trivially $s_m \to 1$, $s_m \to 2$) of the survivor matrices at time $m + 1$.

4. The two-to-two transitions ($1 \to 2$, $2 \to 1$ or $1 \to 1$, $2 \to 2$) of survivor matrices for all time instants l, $m + 2 \leq l \leq k$.

5. The edges corresponding to all the transitions in 3 and 4.

In the following, we will first create a partition J^k of the 2-dimensional space $\mathbb{C}$, such that within the same subset of the partition, descriptions 4 and 5 remain fixed for all c. In other words, within each subset of partition J^k the number of different groups is at most equal to the number of all possible combinations of descriptions 1 and 2 (since description 3 is trivial), which is $2k - 1$. This shows that the number of groups α_k at time k is at most $(2k - 1)|J^k|$. Then, We will complete the proof by showing that $|J^k|$ is polynomial in k.

Actually, J^k is created by two sets of partitioning lines, which result in two partitions A^k and B^k, respectively. We will discuss them separately in the following two subsections.

D.3.1 Construction of A^k

Let $\mathcal{E}^{i\to j} \subset \mathcal{E}$ be the set of edges going from state i to state j and

$$\hat{e}_l^{i\to j}(c) \triangleq \arg\max_{e\in\mathcal{E}^{i\to j}} L_l(e,c) \tag{D.6}$$

be the most likely edge going from state i to state j for a given c at time l. For any given i, j and l, consider partition $A_l^{i\to j}$ of the 2-dimensional space $\mathbb{C}$ created by the following partitioning lines

$$L_l(e,c) = L_l(f,c),\ \forall e \neq f,\ e,f \in \mathcal{E}^{i\to j} \tag{D.7}$$

Since each line in (D.7) defines a boundary of two possible results of a pairwise comparison in set $\mathcal{E}^{i\to j}$, within each bounded area of the partition $A_l^{i\to j}$, $\hat{e}_l^{i\to j}(c)$ defined by all pairwise comparisons in set $\mathcal{E}^{i\to j}$, remains fixed for all c. Now consider the finer partition A^k created by all the lines in (D.7) $\forall i,j \in \mathcal{S}$, $1 \leq l \leq k$. It follows directly from the above discussion that $\hat{e}_l^{i\to j}(c)$ remains fixed for all c in the same subset of partition A^k, $\forall i,j \in \mathcal{S}$, $1 \leq l \leq k$. In other words, description 5 is the same for all survivor matrices at time k in the same subset of partition A^k.

D.3.2 Construction of B^k

Let another set of partitioning lines

$$L_l(e^{1\to 2},c) + L_l(e^{2\to 1},c) = L_l(e^{1\to 1},c) + L_l(e^{2\to 2},c)\ \forall e^{i\to j} \in \mathcal{E}^{i\to j},\ \forall i,j \in \mathcal{S},\ l \leq k \tag{D.8}$$

define partition B_l. Then for any pairs c and c' in the same subset of B_l, the corresponding $\hat{\mathbf{V}}^k(c)$ and $\hat{\mathbf{V}}^k(c')$ can not have different two-to-two transitions ($1 \to 2$,

$2 \to 1$ or $1 \to 1$, $2 \to 2$) at time l, i.e.,

$$\hat{\mathbf{V}}^k(1,l|c) = e^{1\to 2} \in \mathcal{E}^{1\to 2}, \hat{\mathbf{V}}^k(2,l|c) = e^{2\to 1} \in \mathcal{E}^{2\to 1} \quad \text{and} \tag{D.9}$$

$$\hat{\mathbf{V}}^k(1,l|c') = e^{1\to 1} \in \mathcal{E}^{1\to 1}, \hat{\mathbf{V}}^k(2,l|c') = e^{2\to 2} \in \mathcal{E}^{2\to 2} \tag{D.10}$$

can not both be true, because

$$\begin{aligned}
&\hat{\mathbf{V}}^k(1,l|c) = e^{1\to 2} \in \mathcal{E}^{1\to 2}, \hat{\mathbf{V}}^k(2,l|c) = e^{2\to 1} \in \mathcal{E}^{2\to 1} \\
\Rightarrow &\sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(1,i|c), c\right) + L_l(e^{1\to 2}, c) \geq \sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(2,i|c), c\right) + L_l(e^{2\to 2}, c), \quad \text{and} \\
&\sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(2,i|c), c\right) + L_l(e^{2\to 1}, c) \geq \sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(1,i|c), c\right) + L_l(e^{1\to 1}, c) \\
\Rightarrow &L_l(e^{1\to 2}, c) + L_l(e^{2\to 1}, c) \geq L_l(e^{2\to 2}, c) + L_l(e^{1\to 1}, c)
\end{aligned} \tag{D.11}$$

and

$$\begin{aligned}
&\hat{\mathbf{V}}^k(1,l|c') = e^{1\to 1} \in \mathcal{E}^{1\to 1}, \hat{\mathbf{V}}^k(2,l|c') = e^{2\to 2} \in \mathcal{E}^{2\to 2} \\
\Rightarrow &\sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(1,i|c'), c'\right) + L_l(e^{1\to 1}, c') \geq \sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(2,i|c'), c'\right) + L_l(e^{2\to 1}, c'), \quad \text{and} \\
&\sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(2,i|c'), c'\right) + L_l(e^{2\to 2}, c') \geq \sum_{i=1}^{l-1} L_i\left(\hat{\mathbf{V}}^k(1,i|c'), c'\right) + L_l(e^{1\to 2}, c') \\
\Rightarrow &L_l(e^{1\to 1}, c') + L_l(e^{2\to 2}, c') \geq L_l(e^{2\to 1}, c') + L_l(e^{1\to 2}, c')
\end{aligned} \tag{D.12}$$

contradict with the fact that c and c' are in the same subset of B_l. Therefore for all c in the same subset of B_l the two-to-two transitions at time $l \leq k$ for all survivor matrices $\hat{\mathbf{V}}^k(c)$ are the same. If we construct the finer partition B^k by intersecting all the lines in (D.8) $\forall 1 \leq l \leq k$, then for all c in the same subset of B^k, the two-to-two transitions of survivor matrices $\hat{\mathbf{V}}^k(c)$ at all time instants l, $1 \leq l \leq k$ are fixed, i.e.,

description 4 is fixed for all survivor matrices at time k for all c in the same subset of B^k.

Now, we construct J^k by intersecting all subsets of A^k and B^k. It follows from the above discussion that descriptions 4 and 5 remain fixed for all survivors matrics for all c in the same subset of partition J^k.

To enumerate the size of $|J^k|$, first observe that $A_l^{i\to j}$ is a partition created by at most $\frac{K(K-1)}{2}$ lines $\forall i,j \in \{1,2\}$ $\forall 1 \leq l \leq k$. Therefore A^k is a partition created by at most $2^2 k\frac{K(K-1)}{2}$ lines. Moreover, since B_l is created by at most $(\frac{K}{2})^4$ lines $\forall 1 \leq l \leq k$, B^k is created by at most $(\frac{K}{2})^4 k$ lines. Altogether, we conclude that J^k is a partition created by at most $2^2 k\frac{K(K-1)}{2} + (\frac{K}{2})^4 k$ lines, whose size is polynomial in k [79]. This completes the proof of Lemma 6.3.

APPENDIX E

Proof of Lemma 7.1 in Chapter 7

Define sequentially

(i) $b_{ij}^{(l)}$ = number of combinations of different i nonzero codewords under one topmost **check** node in $\mathcal{C}_r$ such that the Hamming weight of the OR of these i codewords is j,

(ii) $B_i^{(l)}(x) = \sum_{j=1}^{\infty} b_{ij}^{(l)} x^j$, and

(iii) $Z_l(x) = \sum_{i=1}^{\infty} (-1)^{i-1} Z_i^{(l)}(x)$.

We first prove the following lemma.

Lemma E.1

$$Z_l(x) = \binom{d_c-1}{1} N_{l-1}(x) - \binom{d_c-1}{2} N_{l-1}^2(x) + \cdots + (-1)^{d_c-2} \binom{d_c-1}{d_c-1} N_{l-1}^{d_c-1}(x) \tag{E.1a}$$

$$= 1 - (1 - N_{l-1}(x))^{d_c-1}. \tag{E.1b}$$

Proof: Consider "OR"ing i different nonzero codewords under one particular check node at the topmost level of the tree of level-$(l+1)$. Recall that in each

such codeword, there is exactly one level-l subtree having root bit equal to 1, while the other $d_c - 2$ subtrees are identically zero. Let j denote the number of level-l subtrees each of whose root bit is equal to 1 in at least one of the i codewords, and let $d_c - 1 - j$ denote the number of level-l subtrees whose root bits are equal to 0 in all the i codewords. We will show that each term of $Z_l(x)$ in (E.1a) corresponds to each j taking values from 1 to $d_c - 1$.

Indeed for $j = 1$, the only contribution to the OR of the i codewords is coming from the single level-l subtree with root 1, and thus the contribution is $A_i^{(l-1)}(x)$. In other words, the part of $B_i^{(l)}(x)$ corresponding to $j = 1$ is exactly $\binom{d_c-1}{1} A_i^{(l-1)}(x)$, and thus, the corresponding contribution to $Z_l(x)$ is exactly

$$\binom{d_c - 1}{1} \sum_{i=1}^{\infty} (-1)^{i-1} A_i^{(l-1)}(x) = \binom{d_c - 1}{1} N_{l-1}(x), \tag{E.2}$$

where the factor $\binom{d_c-1}{1}$ enumerates the ways to pick a subtree among $d_c - 1$ ones.

Now consider the case of $j = 2$. There are $\binom{d_c-1}{2}$ ways to pick these two subtrees that have non-zero contributions to the OR of the i codewords. Let the first subtree contributes i_1 non-zero codewords and the second contributes i_2 non-zero codewords. Then, the corresponding individual contributions for each subtree are $A_{i_1}^{(l-1)}(x)$ and $A_{i_2}^{(l-1)}(x)$, respectively. The total contribution to the OR of the i codewords is thus $\sum_{i_1+i_2=i} A_{i_1}^{(l-1)}(x) A_{i_2}^{(l-1)}(x)$. In other words, the part of $B_i^{(l)}(x)$ corresponding to $j = 2$ is exactly $\binom{d_c-1}{2} \sum_{i_1+i_2=i} A_{i_1}^{(l-1)}(x) A_{i_2}^{(l-1)}(x)$. As a result, the total contribution

to $Z_l(x)$ corresponding to $j = 2$ is

$$\begin{aligned}
&\binom{d_c-1}{2}\sum_{i=2}^{\infty}(-1)^{i-1}\sum_{i_1+i_2=i}A_{i_1}^{(l-1)}(x)A_{i_2}^{(l-1)}(x)\\
&=-\binom{d_c-1}{2}\sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}(-1)^{i_1+i_2-2}A_{i_1}^{(l-1)}(x)A_{i_2}^{(l-1)}(x)\\
&=-\binom{d_c-1}{2}\left[\sum_{i_1=1}^{\infty}(-1)^{i_1-1}A_{i_1}^{(l-1)}(x)\right]^2\\
&=-\binom{d_c-1}{2}N_{l-1}^2(x). \qquad \text{(E.3)}
\end{aligned}$$

Hence, the lemma is true for $d_c = 2$ and 3. Now, the general statement of the lemma follows from induction on d_c. ■

Next, we are going to prove that

$$N_l(x) = x[Z_l(x)]^{d_v-1}. \qquad \text{(E.4)}$$

To prove this equation, it is sufficient to prove the following lemma, where we call $N_l(x)$ the *code distribution* of $\mathcal{C}_r$.

Lemma E.2 *Consider a code $\mathcal{C}$ consisting of two component codes $\mathcal{C}_1$ and $\mathcal{C}_2$, such that*

$$\mathcal{C} = \{[\mathbf{c}_1, \mathbf{c}_2] : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}. \qquad \text{(E.5)}$$

Let $Q(x) = \sum_i (-1)^{i-1} q_i(x)$ and $R(x) = \sum_i (-1)^{i-1} r_i(x)$ be the code distributions of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Then, the code distribution of $\mathcal{C}$ is $Q(x)R(x)$.

Proof: For any set of i distinct codewords

$$\mathcal{S} = \{[\mathbf{c}_{11}, \mathbf{c}_{21}], [\mathbf{c}_{12}, \mathbf{c}_{22}], \ldots, [\mathbf{c}_{1i}, \mathbf{c}_{2i}]\} \subset \mathcal{C}, \qquad \text{(E.6)}$$

define the projection functions

$$P_1(\mathcal{S}) \triangleq \{\mathbf{c}_1 \in \mathcal{C}_1 : \exists \mathbf{c}_2 \in \mathcal{C}_2, \text{such that } [\mathbf{c}_1, \mathbf{c}_2] \in \mathcal{S}\} \tag{E.7a}$$

$$P_2(\mathcal{S}) \triangleq \{\mathbf{c}_2 \in \mathcal{C}_2 : \exists \mathbf{c}_1 \in \mathcal{C}_1, \text{such that } [\mathbf{c}_1, \mathbf{c}_2] \in \mathcal{S}\}. \tag{E.7b}$$

Now, we choose a set of j codewords $\mathcal{B}_{1j}$ from $\mathcal{C}_1$ and a set of k codewords $\mathcal{B}_{2k}$ from $\mathcal{C}_2$. Let $y_i^{(j,k)}$ be the number of combinations of i different codewords in $\mathcal{C}$, such that the set of these i codewords $\mathcal{S}$ satisfies $P_1(\mathcal{S}) = \mathcal{B}_{1j}$ and $P_2(\mathcal{S}) = \mathcal{B}_{2k}$. If i is even (odd, respectively), then this combination will contribute a negative (positive) term in the code distribution of $\mathcal{C}$. Hence, we can define $Y^{(j,k)} \triangleq \sum_{i=1}^{\infty}(-1)^{i-1} y_i^{(j,k)}$ to be the summed up term for all sets of codewords whose projection on $\mathcal{C}_1$ equals $\mathcal{B}_{1j}$, and projection on $\mathcal{C}_2$ equals $\mathcal{B}_{2k}$. In the following, we will use induction to show that $Y^{(j,k)} = (-1)^{j+k}$.

When $j = k = 1$, it is clear that $Y^{(1,1)} = 1$. Suppose $Y^{(j',k')} = (-1)^{j'+k'}$. When $j = j' + 1$ and $k = k'$, let $\mathbf{c}_1$ be some codeword in $\mathcal{B}_{1(j'+1)}$. For any set of codewords $\mathcal{S}$, if $P_1(\mathcal{S}) = \mathcal{B}_{1(j'+1)}$ and $P_2(\mathcal{S}) = \mathcal{B}_{2k'}$, then $\mathcal{S}$ must include at least one codeword in the set $\mathcal{A} \triangleq \{[\mathbf{c}_1, \mathbf{c}_2] : \mathbf{c}_2 \in \mathcal{B}_{2k'}\}$. Counting the number of combinations of $i = g + h$ codewords, whose g codewords form a set $\mathcal{S}_1$ satisfying $P_1(\mathcal{S}_1) = \mathcal{B}_{1(j'+1)} \setminus \{\mathbf{c}_1\}$ (note that $\mathcal{B}_{1(j'+1)} \setminus \{\mathbf{c}_1\}$ is some $\mathcal{B}_{1j'}$) and $P_2(\mathcal{S}_2) = \mathcal{B}_{2k'}$, and the other h codewords form a set $\mathcal{S}_2$ satisfying $\mathcal{S}_2 \subset \mathcal{A}$, we have

$$y_i^{(j'+1,k')} = \sum_{h=1}^{k'} \binom{k'}{h} y_{i-h}^{(j',k')}, \quad \forall i \tag{E.8}$$

since there are $\binom{k'}{h}$ ways of choosing h codewords in $\mathcal{B}_{2k'}$ to be paired with $\mathbf{c}_1$ in $\mathcal{A}$. This implies that

$$\begin{aligned}
Y^{(j'+1,k')} =& \sum_{i=1}^{\infty}(-1)^{i-1}y_i^{(j'+1,k')} && \text{(E.9a)}\\
=& \sum_{i=1}^{\infty}(-1)^{i-1}\sum_{h=1}^{k'}\binom{k'}{h}y_{i-h}^{(j',k')} && \text{(E.9b)}\\
=& \sum_{g=1}^{\infty}\sum_{h=1}^{k'}(-1)^{g+h-1}\binom{k'}{h}y_g^{(j',k')} && \text{(E.9c)}\\
=& \sum_{g=1}^{\infty}(-1)^{g-1}y_g^{(j',k')}\sum_{h=1}^{k'}(-1)^h\binom{k'}{h} && \text{(E.9d)}\\
=& Y^{(j',k')}[(1-x)^{k'}-1]|_{x=1} && \text{(E.9e)}\\
=& Y^{(j',k')}(-1) && \text{(E.9f)}\\
=& (-1)^{j'+1+k'}, && \text{(E.9g)}
\end{aligned}$$

where we have used the fact that $y_g^{(j',k')} = 0$ for all non-positive g. A similar argument can be used to show that $Y^{(j',k'+1)} = (-1)^{j'+k'+1}$. So, by induction we have that $Y^{(j,k)} = (-1)^{j+k}$.

Since the number of ways of forming distinct pairs $\{\mathcal{B}_{1j}, \mathcal{B}_{2k}\}$ is captured by the coefficients of $q_j(x)r_k(x)$, and the associated weight of the OR of every set of codewords $\mathcal{S}$ such that $P_1(\mathcal{S}) = \mathcal{B}_{1j}$ and $P_2(\mathcal{S}) = \mathcal{B}_{2k}$ is captured by the powers of x of the corresponding term in $q_j(x)r_k(x)$, it follows that the code distribution of $\mathcal{C}$ is

$$\sum_{j,k} Y^{(j,k)}q_j(x)r_k(x) = \sum_{j,k}(-1)^{j+k-2}q_j(x)r_k(x) = Q(x)R(x). \qquad \text{(E.10)}$$

■

Now, since there are $d_v - 1$ component codes emanating from the root bit, and the root bit is always 1 for nonzero codewords, we have

$$N_l(x) = x[Z_l(x)]^{d_v-1} = x[1 - (1 - N_{l-1}(x))^{d_c-1}]^{d_v-1}, \tag{E.11}$$

which completes the proof of the whole lemma.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, pp. 379–423, July 1948.

[2] N. Shulman and M. Feder, "Random coding techniques for nonrandom codes," *IEEE Trans. Information Theory*, vol. 45, no. 6, pp. 2101–2104, Sept. 1999.

[3] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[4] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 2nd edition, 1993.

[5] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," in *Proc. ACM Symposium on Theory of Computing*, Dallas, Texas, May 1998, pp. 10–19.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. International Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[8] N. Wiberg, *Codes and Decoding on General Graphs*, Ph.D. thesis, Linköping University, Linköping, Sweden, 1996.

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

[10] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. ACM Symposium on Theory of Computing*, El Paso, Texas, May 1997, pp. 150–159.

[11] M. A. Shokrollahi, "New sequences of linear time erasure codes approaching channel capacity," in *Proc. International Symposium on Information Theory and its Applications*, Honolulu, Hawaii, Nov. 1999, pp. 65–76.

[12] A. Shokrollahi, "Capacity-achieving sequences," in *Codes, Systems, and Graphical Models*, B. Marcus and J. Rosenthal, Eds., vol. 123 of *IMA Volumes in Mathematics and its Applications*, pp. 153–166. Springer-Verlag, 2000.

[13] P. Oswald and A. Shokrollahi, "Cappacity-achieving sequences for the erasure channel," *IEEE Trans. Information Theory*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

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

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

[16] T. Richardson and R. Urbanke, "Multi-edge type LDPC codes," 2004, [Online]. Available: http://lthcwww.epfl.ch/papers/multiedge.ps.

[17] H. Jin, A. Khandekar, and R. J. McEliece, "Irregular repeat-accumulate codes," in *Proc. International Symposium on Turbo Codes and Related Topics*, Brest, France, Sept. 2000, pp. 1–8.

[18] H. D. Pfister, I. Sason, and R. Urbanke, "Capacity-achieving ensembles for the binary erasure channel with bounded complexity," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2352–2379, July 2005.

[19] A. Barg and G. Zemor, "Error exponents of expander codes," *IEEE Trans. Information Theory*, vol. 48, no. 6, pp. 1725–1729, June 2002.

[20] A. Khandekar and R. J. McEliece, "On the complexity of reliable communication on the erasure channel," in *Proc. International Symposium on Information Theory*, Washinton, DC, June 2001, p. 1.

[21] A. Khandekar, *Graph-based Codes and Iterative Decoding*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2002.

[22] N. Varnica and M. Fossorier, "Belief-propagation with information correction: improved near maximum-likelihood decoding of low-density parity-check codes," in *Proc. International Symposium on Information Theory*, Chicago, USA, June 2004, p. 343.

[23] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *IEEE Trans. Information Theory*, vol. 50, no. 3, pp. 439–454, Mar. 2004.

[24] D. Burshtein and G. Miller, "An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Information Theory*, vol. 50, no. 11, pp. 2837–2844, Nov. 2004.

[25] M. Lentmaier, D. Truhachev, K. Zigangirov, and D. Costello, "An analysis of the block error probability performance of iterative decoding," *IEEE Commun. Lett.*, vol. 9, no. 12, pp. 1067–1069, Dec. 2005.

[26] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. Allerton Conf. Commun., Control, Comp.*, Illinois, Sept. 1998, pp. 201–210.

[27] I. Sason and R. Urbanke, "Parity-check density versus performance of binary linear block codes over memoryless symmetric channels," *IEEE Trans. Information Theory*, vol. 49, no. 7, pp. 1611–1635, July 2003.

[28] D. Divsalar, "A simple tight bound on error probability of block codes with application to turbo codes," Tech. Rep. TMO Progress Report 42-139, Jet Propulsion Labs., Pasadena, CA, Nov. 1999.

[29] J. Garcia-Frias and W. Zhong, "Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 266–268, June 2003.

[30] H. Jin and T. Richardson, "Block error iterative decoding capacity for LDPC codes," in *Proc. International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 52–56.

[31] I. Motedayen and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. Communications*, vol. 51, no. 2, pp. 197–207, Feb. 2003.

[32] R. J. McEliece and W. E. Stark, "Channels with block interference," *IEEE Trans. Information Theory*, vol. 30, no. 1, pp. 44–53, Jan. 1984.

[33] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley & Sons, New York, NY, 1968.

[34] C.-H. Hsu and A. Anastasopoulos, "Subexponential-complexity exact sequence detection in the presence of frequency and phase uncertainty," in *Proc. International Conf. Communications*, Paris, France, June 2004, pp. 20–24.

[35] C.-H. Hsu and A. Anastasopoulos, "Maximum likelihood decoding of trellis codes in fading channels with no receiver CSI is a polynomial-complexity problem," in *Proc. International Symposium on Information Theory*, Chicago, IL, July 2004, p. 147.

[36] C.-H. Hsu and A. Anastasopoulos, "Design and analysis of joint data detection and frequency/phase estimation algorithms," *IEEE J. Select. Areas Commun.*, vol. 23, no. 9, pp. 1707–1717, Sept. 2005.

[37] C.-H. Hsu and A. Anastasopoulos, "Capacity achieving LDPC codes through puncturing," in *Proc. International Conf. on Wireless Networks, Commun., and Mobile Comp.*, Maui, Hawaii, June 2005, pp. 1575–1580.

[38] C.-H. Hsu and A. Anastasopoulos, "Asymptotic weight distributions of irregular repeat-accumulate codes," in *Proc. Global Telecommunications Conference*, St. Louis, MO, Nov. 2005, pp. 1147–1151.

[39] C.-H. Hsu and A. Anastasopoulos, "Capacity-achieving codes with bounded graphical complexity on noisy channels," in *Proc. Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2005.

[40] C.-H. Hsu and A. Anastasopoulos, "Capacity-achieving codes with bounded graphical complexity and maximum likelihood decoding," *IEEE Trans. Information Theory*, Mar. 2006, (Submitted).

[41] C.-H. Hsu and A. Anastasopoulos, "Iterative decoding performance bounds for LDPC codes on noisy channels," 2006, (In preparation).

[42] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[43] T. J. Richardson S.-Y. Chung and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[44] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes," in *Proc. Information Theory Workshop*, San Antonio, USA, Oct. 2004.

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

[46] O. Pretzel, *Error-Correcting Codes and Finite Fields*, Oxford University Press, New York, 1992.

[47] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. Information Theory*, vol. 40, no. 4, pp. 1284–1292, July 1994.

[48] G. Miller and D. Burshtein, "Bounds on the maximum-likelihood decoding error probability of low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2696–2710, Nov. 2001.

[49] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[50] A. Brown, M. Luby, and A. Shokrollahi, "Repeat-accumulate codes that approach the Gilbert-Varshamov bound," in *Proc. International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 169–173.

[51] A. Barg and G. Zemor, "Distance properties of expander codes," *IEEE Trans. Information Theory*, vol. 52, no. 1, pp. 78–90, Jan. 2006.

[52] H. D. Pfister and I. Sason, "Accumulate-repeat-accumulate codes: Systematic codes achieving the binary erasure channel with bounded complexity," in *Proc. Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2005, [Online]. Available: http://www.arxiv.org/abs/cs.IT/0509044.

[53] I. Sason, E. Telatar, and R. Urbanke, "On the asymptotic input-output weight distributions and thresholds of convolutional and turbo-like encoders," *IEEE Trans. Information Theory*, vol. 48, no. 12, pp. 3052–3061, Dec. 2002.

[54] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate repeat accumulate codes," in *Proc. International Symposium on Information Theory*, Chicago, USA, June 2004, p. 505.

[55] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Information Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[56] D. Burshtein and G. Miller, "Asymptotic enumeration methods for analyzing LDPC codes," *IEEE Trans. Information Theory*, vol. 50, no. 6, pp. 1115–1131, June 2004.

[57] L. M. J. Bazzi and S. K. Mitter, "Encoding complexity versus minimum distance," *IEEE Trans. Information Theory*, vol. 51, no. 6, pp. 2103–2112, June 2005.

[58] S. Shamai and I. Sason, "Variations on the gallager bounds, connections, and applications," *IEEE Trans. Information Theory*, vol. 48, no. 12, pp. 3029–3051, Dec. 2002.

[59] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[60] H. Pishro-Nik, N. Rahnavard, and F. Fekri, "Nonuniform error correction using low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2702–2714, July 2005.

[61] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Communications*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[62] C. Measson, A. Montanari, and R. Urbanke, "Why we can not surpass capacity: The matching condition," in *Proc. Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2005.

[63] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge University Press, Cambridge, UK, 2006.

[64] R. J. McEliece, *The Theory of Information and Coding, 2nd ed.*, Cambridge University Press, Cambridge, UK, 2002.

[65] M. Morelli and U. Mengali, "Feedforward frequency estimation for PSK: a tutorial review," *European Trans. Telecommun.*, vol. 9, no. 2, pp. 103–116, March/April 1998.

[66] C. N. Georghiades and D. L. Snyder, "The expectation-maximization algorithm for symbol unsynchronized sequence detection," *IEEE Trans. Communications*, vol. 39, no. 1, pp. 54–61, Jan. 1991.

[67] M. Ghosh and C. L. Weber, "Maximum-likelihood blind equalization," in *Proc. SPIE*, July 1991, pp. 181–195.

[68] C. N. Georghiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Communications*, vol. 45, no. 3, pp. 300–308, Mar. 1997.

[69] S. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. Communications*, vol. 38, no. 1, pp. 3–12, Jan. 1990.

[70] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Communications*, vol. 32, no. 2, pp. 169–176, Feb. 1984.

[71] J. Lodge and M. Moher, "Maximum likelihood estimation of CPM signals transmitted over Rayleigh flat fading channels," *IEEE Trans. Communications*, vol. 38, no. 6, pp. 787–794, June 1990.

[72] R. A. Iltis, "A Bayesian maximum-likelihood sequence estimation algorithm for a priori unknown channels and symbol timing," *IEEE J. Select. Areas Commun.*, vol. 10, no. 3, pp. 579–588, Apr. 1992.

[73] A. N. D'Andrea, U. Mengali, and G. M. Vitetta, "Approximate ML decoding of coded PSK with no explicit carrier phase reference," *IEEE Trans. Communications*, vol. 42, no. 2/3/4, pp. 1033–1039, Feb./Mar./April 1994.

[74] X. Yu and S. Pasupathy, "Innovations-based MLSE for Rayleigh fading channels," *IEEE Trans. Communications*, vol. 43, no. 2/3/4, pp. 1534–1544, Feb./Mar./Apr. 1995.

[75] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Communications*, vol. 43, no. 2/3/4, pp. 354–364, Feb/Mar/Apr. 1995.

[76] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, 4th edition, 2001.

[77] K. M. Mackenthun, Jr., "A fast algorithm for multiple-symbol differential detection of MPSK," *IEEE Trans. Communications*, vol. 42, no. 2/3/4, pp. 1471–1474, Feb./Mar./Apr. 1994.

[78] W. Sweldens, "Fast block noncoherent decoding," *IEEE Commun. Lett.*, vol. 5, no. 4, pp. 132–134, Apr. 2001.

[79] H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *Society for Industrial and Applied Mathematics Journal on Computing*, vol. 15, pp. 341–363, 1986.

[80] H. Leib and S. Pasupathy, "The phase of a vector perturbed by Gaussian noise and differentially coherent receivers," *IEEE Trans. Information Theory*, vol. 34, no. 6, pp. 1491–1501, Nov. 1988.

[81] R. Nuriyev and A. Anastasopoulos, "Pilot-symbol-assisted coded transmission over the block-noncoherent AWGN channel," *IEEE Trans. Communications*, vol. 51, no. 6, pp. 953–963, June 2003.

[82] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Trans. Communications*, vol. 43, no. 2/3/4, pp. 1169–1178, Feb./Mar./Apr. 1995.

[83] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[84] I. Motedayen and A. Anastasopoulos, "Polynomial complexity ML sequence and symbol-by-symbol detection in fading channels," in *Proc. International Conf. Communications*, Anchorage, Alaska, May 2003, pp. 2718–2722.

[85] B. J. Frey, *Graphical models for machine learning and digital communications*, MIT Press, Cambridge, MA, 1998.

[86] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *European Trans. Telecommun.*, vol. 9, no. 2, pp. 155–172, March/April 1998.

[87] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic Publishers, 2001.

[88] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, John Wiley & Sons, New York, NY, 1998.

[89] C. Di, D. Proietti, E. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Information Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

ABSTRACT

Design and Analysis of Capacity-Achieving Codes and Optimal Receivers with Low Complexity

by

Chun-Hao Hsu

Chair: Achilleas Anastasopoulos

High performance channel coding schemes for digital communication systems with low computational complexity are considered under two scenarios. Firstly, when the channel is a memoryless binary-input output-symmetric (MBIOS) channel, we design and analyze channel codes defined on factor graphs that can operate reliably and arbitrarily close to the channel capacity.

We show that punctured low-density parity-check (LDPC) codes can achieve capacity on any MBIOS channel under maximum-likelihood (ML) decoding. Moreover, we prove that puncturing preserves the multiplicative gap to capacity of the original LDPC codes with a small enough rate, a fact that suggests high potential of using punctured LDPC codes in rate-compatible coding.

The open problem of whether codes with state nodes can achieve capacity on general MBIOS channels with bounded graphical complexity is investigated next. Nonsystematic irregular repeat-accumulate (NIRA) codes are viewed as potential

candidates and their ML decoding performance is analyzed. Unable to assert the capacity-achieving capability of NIRA codes, we propose a new family of codes, namely the low-density parity-check and generator matrix codes, and prove that they can achieve capacity on any MBIOS channel using ML decoding and on any binary erasure channel (BEC) using belief propagation decoding with bounded graphical complexity.

Motivated by the need to analytically characterize the iterative decoding performance of codes on general MBIOS channels, we give tight performance lower and upper bounds for LDPC codes with iterative decoding. Furthermore, we use these bounds to show that multi-edge type LDPC codes, including nearly all known codes defined on graphs, have the best iterative decoding performance on the BEC among all MBIOS channels with the same uncoded bit error probability, and similarly, have the worst iterative decoding performance on the BEC among all MBIOS channels with the same Bhattacharyya parameter.

In the second scenario, more complicated channels with memory are considered. We focus on the channel with additive white Gaussian noise (AWGN) and frequence/phase-jitter, and the flat-fading channel under the block independent assumption to characterize the channel dynamics. We propose polynomial complexity algorithms for optimal detection/decoding without channel state information at the receiver. This result challenges the traditionally believed exponential complexity demand for both uncoded and two-state trellis-coded transmissions.