Deterministic Design and Performance Analysis of Low-Density Parity-Check Codes in the Presence of Channel Estimation Error

by

Hamid Saeedi

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

OTTAWA-CARLETON INSTITUTE FOR ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

DEPARTMENT OF SYSTEMS AND COMPUTER ENGINEERING

CARLETON UNIVERSITY

OTTAWA, ONTARIO

August 2007

Copyright © Hamid Saeedi, 2007
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Canada
Abstract

The Belief Propagation (BP) algorithm for decoding of low-density parity-check (LDPC) codes over a binary input additive white Gaussian noise channel (BIAWGN) requires the knowledge of the signal-to-noise ratio (SNR) at the receiver to achieve its ultimate performance. An erroneous estimation or the absence of perfect knowledge of the SNR is referred to as “SNR mismatch”. SNR mismatch can significantly degrade the performance of LDPC codes decoded by the BP algorithm.

In this doctoral thesis, we investigate the performance and design of low-density parity-check (LDPC) codes in the presence of mismatch. We show that at the extremes for over- and underestimation of SNR, the performance of belief propagation tends to that of min-sum algorithm and the channel bit error rate, respectively. The asymptotic analysis for regular codes indicates that the sensitivity to mismatch increases by increasing the variable node degree and by decreasing the check node degree. The effect of variable node degree however is more profound such that at a given rate, the codes with the smallest variable and check degrees are more robust against SNR mismatch. For irregular codes, we demonstrate that the ensembles conventionally designed and optimized assuming perfect channel estimation, can perform worse in the presence of mismatch. Using extrinsic information transfer (EXIT) charts, we design irregular LDPC codes that perform well both in the presence and absence of mismatch.

The existing methods for code design are based on the application of asymptomatic analysis tools such as density evolution or EXIT charts in an optimization process. This approach is usually very time-consuming and computationally complex. In this thesis, we propose a deterministic approach to design LDPC codes in both the
presence and absence of mismatch. The design complexity of this approach is far less than that of conventional methods. We demonstrate that LDPC codes that are designed by the proposed approach perform very closely to the best codes designed by optimization. We initially develop a method to deterministically design LDPC codes for a simple channel model called the Binary Erasure Channel (BEC). We then extend this method to BIAWGN channels.

As an important result for BEC, we show that the performance of the codes designed based on our method can achieve the capacity of BEC when the maximum variable node and check node degrees tend to infinity. Based on this fact, we introduce new sequences of capacity achieving LDPC code ensembles over the BEC. Moreover, we prove that the proposed sequences are asymptotically quasi-optimal, i.e., their decoding complexity per iteration increases only logarithmically with the relative increase of the threshold. We also show that with proper choice of the sequence parameters, they can be made asymptotically optimal, i.e., their decoding complexity per iteration increases not only logarithmically with the relative increase of the performance but also the coefficient of this increase, is equal to one.
Acknowledgement

I would like to express my deepest gratitude to my advisor, Dr. Amir H. Banihashemi for his continuous support and encouragement throughout my Ph.D. program. He always welcomed new ideas and I benefited much from his knowledge, excellent suggestions, editorial skills, and passion for doing original research.

I am grateful to Dr. Daniel J. Costello, Jr., from the Department of Electrical Engineering, University of Notre Dame to serve as the external examiner of my Ph.D. defense. I would also like to thank other committee members of my defense: Dr. David Falconer, Dr. Ian Marsland, and Dr. Calvin Plett, from our university and Dr. Ali Miri from University of Ottawa.

My research work was supported by grants and awards provided by Natural Science and Engineering Research Council, Ontario Graduate Scholarship in Science and Technology and Ontario Graduate Scholarship programs, and Carleton University.

I am thankful to other faculty members of Carleton University, in particular Dr. Richard M. Dansereau. I would also like to appreciate the efforts of the department staff. My special thanks go to Ms. Hebert, Ms. Power, Ms. Bowman, Ms. Lee and Mr. Mehta. I like to thank all my friends and colleagues at the department. I am indebted to my M.Sc. supervisor Dr. Farokh Marvasti and other faculty members of Sharif University of Technology as well as my school teachers.

Last but not least I would like to thank my dear parents, brothers and my beloved wife who also helped me a lot in finalizing this thesis, for their unconditional love and support. I would also like to commemorate my beloved grandmother who passed away few months before my defense. I wish I could share this happy moment with her.
# Table of Contents

List of Tables ................................................................................................................................. ix
List of Figures ........................................................................................................................................ x
List of Appendices ............................................................................................................................ xii
List of Acronyms ................................................................................................................................ xiii
List of Symbols ..................................................................................................................................... xiv

Chapter 1  Introduction ....................................................................................................................... 1

Chapter 2 Background review of Low Density Parity Check codes .............................................. 6

2.1 Digital communication systems ................................................................................................... 6

2.2 Channel Coding ............................................................................................................................ 8

   2.2.1 Linear block codes .............................................................................................................. 8

   2.2.2 Maximum Likelihood (ML) sequence decoding ................................................................. 9

   2.2.3 Bit level Maximum A Posteriori (MAP) decoding ............................................................. 9

2.3 Low-Density Parity-Check Codes ............................................................................................... 10

   2.3.1 LDPC code structure .......................................................................................................... 10

   2.3.2 Tanner graph ..................................................................................................................... 11

   2.3.3 Degree distributions ......................................................................................................... 12

   2.3.4 Message passing decoders ............................................................................................... 13

      2.3.4.1 Belief Propagation (Sum Product) Algorithm: ......................................................... 14

      2.3.4.2 Min-Sum Algorithm .................................................................................................. 15

2.4 Asymptotic performance analysis of LDPC codes ................................................................... 15

   2.4.1 Density evolution .............................................................................................................. 15

   2.4.2 Thresholds ......................................................................................................................... 18

   2.4.3 Symmetry (consistency) property ..................................................................................... 18

2.5 Approximations to density evolution ......................................................................................... 19

   2.5.1 Gaussian approximation ................................................................................................. 19

   2.5.2 EXIT charts ..................................................................................................................... 22

Chapter 3 Performance of LDPC codes in the presence of channel estimation error................. 26

3.1 Problem overview ....................................................................................................................... 27

3.2 A simplified density evolution method based on Gaussian approximation ......................... 28

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
6.5 Concluding remarks .......................................................................................................97

Chapter 7 A deterministic approach for designing LDPC codes over BIAWGN channels
........................................................................................................................................................99

7.1 Introduction .......................................................................................................................99
7.2 Upper bounds on variable node degrees over the BIAWGN channel ......................100
7.3 Deterministic design of LDPC codes for a given channel parameter .....................106
7.4 Deterministic design of LDPC codes for a given code rate........................................110
7.5 Deterministic design of LDPC codes for a given code rate in the presence of
mismatch ................................................................................................................................115
7.6 Concluding remarks .......................................................................................................119

Chapter 8 Conclusion and Future Work .............................................................................120

References .................................................................................................................................149
List of Tables

Table 5-1: Type-MB ensembles for $R = 0.5$ and $P = 4$.......................................................... 75  
Table 5-2: Best Type-MB check-regular ensembles for $R = 0.5$ and different values of $P$. .......................................................... 75  
Table 5-3: Results for rate optimization of LDPC code ensembles with $D_c = 5$ and 4 different constituent variable node degrees at channel erasure probability 0.48............. 79  
Table 6-1: The values of $\lambda$, $\mu$ and $\Delta$ for different ensemble sequences and different check ........................................................................................................................................ 95  
Table 6-2: The values of $P$, $D_v$ and $\psi$ for different ensemble sequences with $f(N) = aN + b$ and different....................................................................................................................................... 96  
Table 7-1: Deterministically optimized degree distributions for rate one half ensembles with different maximum variable node degrees................................................................. 116
List of Figures

Figure 2-1: The Tanner graph corresponding to the code given in the Example 2-1. .................................. 12
Figure 2-2: Exchange of mutual information between variable nodes and check nodes in a (3,6) LDPC code ensemble at SNR=1.5dB. ...................................................................................... 25
Figure 3-1: Variance to mean ratio at the output of variable nodes as a function of iteration number for a (3,6)-regular ensemble with SNR offsets $\eta = -2, 0$ and $2$ dB........ 30
Figure 3-2: Mapping $G$; (a) $g_1(m,\sigma)$, (b) $g_2(m,\sigma)$. .............................................................................. 32
Figure 3-3: Normalized threshold values $\Delta a(\eta)$ for a number of $(d_v,d_c)$ ensembles of regular LDPC codes and for the range of SNR offset $\eta$ between -6 and +6dB. .......... 38
Figure 3-4: BER of a randomly constructed (3,6)-regular LDPC code of length $10^5$ at different SNR values in the presence of SNR mismatch. Corresponding mismatch thresholds are also given ........................................................................................................................................ 40
Figure 3-5: SNR threshold values for three rate $\frac{1}{2}$ ensembles of irregular LDPC codes for the range of SNR offset $\eta$ between -6 and +6dB. ................................................................. 41
Figure 3-6: BER of a randomly constructed rate-1/2 irregular LDPC code of length 20,000 with degree distributions $(\lambda_1,\rho_1)$. Corresponding mismatch thresholds are also given........................................................................................................................................ 41
Figure 4-1: BP ( ) and MS (— ) thresholds for 2 irregular codes with maximum variable node degree of 20. .............................................................................................................. 45
Figure 4-2: Degree-3 variable node and degree-6 check node EXIT charts for different offset values at SNR= 4.2dB. .............................................................................................................. 49
Figure 4-3: BP(-) and MS(---) thresholds for different irregular ensembles. ............................. 54
Figure 4-4: BER performance of $C_1$ and $C_2$ for SNR values of 1 dB and 1.3 dB at different offset values. ...................................................................................................................... 56
Figure 4-5: BER performance of $C_1$ and $C_2$ at $\eta=0$. ......................................................... 58
Figure 5-1: The performance of codes with block length 5000 from the ensembles of Example 5-8. .............................................................................................................................. 77
Figure 6-1: The required maximum variable node degree for different performances and different values of $a$. .............................................................................................................. 98

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6-2: The required number of constituent variable node degree for different performances and different values of $a$. ................................................................. 98

Figure 7-1: The values of $\lambda_{3,max}$ versus $d_e$ and $\sigma^2$ the noise variance for BIAWGN channel. .......................................................................................................................... 105

Figure 7-2: BP thresholds for ensembles $C_i$ and $C_{mismatch}$ for different offset values. .... 119
List of Appendices

Appendix A: Proof of the Chapter 5 lemmas and theorems.................................124
Appendix B: Fractional Binominal Coefficients ..................................................132
Appendix C: Proof of the Chapter 6 lemmas and theorems.................................133
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP</td>
<td>A-Posteriori Probability</td>
</tr>
<tr>
<td>BEC</td>
<td>Binary Erasure Channel</td>
</tr>
<tr>
<td>BIAWGN</td>
<td>Binary Input Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BICM-ID</td>
<td>Bit Interleaved Coded Modulation-Iterative Demapping</td>
</tr>
<tr>
<td>BSC</td>
<td>Binary Symmetric Channel</td>
</tr>
<tr>
<td>BP</td>
<td>Belief Propagation</td>
</tr>
<tr>
<td>BCH</td>
<td>Bose, Ray-Chaudhuri, Hocquenghem</td>
</tr>
<tr>
<td>BCJR</td>
<td>Bahl-Cocke-Jelinek-Raviv</td>
</tr>
<tr>
<td>DE</td>
<td>Density Evolution</td>
</tr>
<tr>
<td>EXIT</td>
<td>Extrinsic Information Transfer</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>LDPC</td>
<td>Low Density Parity Check</td>
</tr>
<tr>
<td>LLR</td>
<td>Log-Likelihood Ratio</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MS</td>
<td>Min-Sum</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PMF</td>
<td>Probability Mass Functions</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
</tbody>
</table>
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha(\eta)$</td>
<td>SNR threshold</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$1/(D_c-1)$</td>
</tr>
<tr>
<td>$\beta^{(o)}(s)$</td>
<td>mismatch threshold for overestimation of SNR</td>
</tr>
<tr>
<td>$\beta^{(u)}(s)$</td>
<td>mismatch threshold for underestimation of SNR</td>
</tr>
<tr>
<td>$\Delta \beta(s)$</td>
<td>measure of tolerance</td>
</tr>
<tr>
<td>$C_i$</td>
<td>code ensemble $i$</td>
</tr>
<tr>
<td>$C(.)$</td>
<td>capacity</td>
</tr>
<tr>
<td>$D_v$</td>
<td>maximum variable node degree</td>
</tr>
<tr>
<td>$D_c$</td>
<td>maximum check node degree</td>
</tr>
<tr>
<td>$\bar{d}_c$</td>
<td>average variable node degrees</td>
</tr>
<tr>
<td>$\bar{d}_v$</td>
<td>average check node degrees</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>asymptotical optimality factor</td>
</tr>
<tr>
<td>$\eta$</td>
<td>SNR mismatch ratio</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>channel erasure probability, threshold for codes over BEC</td>
</tr>
<tr>
<td>$3$</td>
<td>upper bound on the achievable threshold</td>
</tr>
<tr>
<td>$E_b$</td>
<td>the average energy per information bit</td>
</tr>
<tr>
<td>$E(\ell)$</td>
<td>erasure probability of messages at iteration $\ell$</td>
</tr>
<tr>
<td>$E{\cdot}$</td>
<td>expectation</td>
</tr>
<tr>
<td>$f_{\text{out}}$</td>
<td>density of the random variable at the check node output</td>
</tr>
<tr>
<td>$f_{v,\ell,j}$</td>
<td>density at the output of the degree $j$ variable node at $\ell$'th iteration</td>
</tr>
<tr>
<td>$f_{c,\ell,k}$</td>
<td>density at the output of the degree $k$ check node at $\ell$'th iteration</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>probability density function</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>check node/variable node operator</td>
</tr>
<tr>
<td>$\psi$</td>
<td>the ratio of threshold to the code rate for BEC</td>
</tr>
<tr>
<td>$G$</td>
<td>generator matrix</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Euler constant</td>
</tr>
<tr>
<td>$H$</td>
<td>parity check matrix</td>
</tr>
<tr>
<td>$I(X;Y)$</td>
<td>The mutual information between $X$ and $Y$</td>
</tr>
<tr>
<td>$I_A$</td>
<td>$I_A$ input mutual information</td>
</tr>
<tr>
<td>$I_{Ev}$</td>
<td>$I_{Ev}$ variable node input mutual information</td>
</tr>
<tr>
<td>$I_{Av}$</td>
<td>$I_{Av}$ variable node output mutual information</td>
</tr>
<tr>
<td>$I_{Ec}$</td>
<td>$I_{Ec}$ check node input mutual information</td>
</tr>
<tr>
<td>$I_{Ac}$</td>
<td>$I_{Ac}$ check node output mutual information</td>
</tr>
<tr>
<td>$I^k_{Ec}$</td>
<td>mutual information at the output of a degree $k$ check node</td>
</tr>
<tr>
<td>$I^j_{Ev}$</td>
<td>mutual information at the output of a degree $j$ variable node</td>
</tr>
<tr>
<td>$\ell$</td>
<td>iteration number</td>
</tr>
<tr>
<td>$\lambda(x)$</td>
<td>check node degree distribution</td>
</tr>
<tr>
<td>$\lambda_{ch}$</td>
<td>bit’s channel LLR</td>
</tr>
<tr>
<td>$m_{in}$</td>
<td>mean of consistent Gaussian variable of incoming extrinsic message to variable node</td>
</tr>
<tr>
<td>$m_{out}$</td>
<td>mean of the variable at variable node output</td>
</tr>
<tr>
<td>$m_{ch}$</td>
<td>mean of the channel LLR</td>
</tr>
<tr>
<td>$m_{out,i}$</td>
<td>mean of the variable node output of degree $i$</td>
</tr>
<tr>
<td>$m_{eq}$</td>
<td>mean of the consistent Gaussian random variable that has the same mutual information</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\mu$</td>
<td>asymptotical quasi-optimality factor</td>
</tr>
<tr>
<td>$N_0$</td>
<td>the one-sided power spectral density</td>
</tr>
<tr>
<td>$N$</td>
<td>maximum variable node degree for Type A ensemble</td>
</tr>
<tr>
<td>$\mathcal{M}(.,.)$</td>
<td>Gaussian distribution</td>
</tr>
<tr>
<td>$p$</td>
<td>probability in BSC</td>
</tr>
<tr>
<td>$P$</td>
<td>number of constituent variable node degrees</td>
</tr>
<tr>
<td>$\rho(x)$</td>
<td>variable node degree distribution</td>
</tr>
<tr>
<td>$R$</td>
<td>code rate</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>upper bound on the achievable code rate</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>noise power in AWGN</td>
</tr>
<tr>
<td>$\hat{\sigma}^2$</td>
<td>estimated noise power in AWGN</td>
</tr>
<tr>
<td>$\sigma^2_{out}$</td>
<td>variance of the Gaussian variable at variable node output</td>
</tr>
<tr>
<td>$\sigma^2_{in}$</td>
<td>variance of the Gaussian variable at variable node output</td>
</tr>
<tr>
<td>$\sigma^2_0$</td>
<td>variance of the Gaussian variable at channel output</td>
</tr>
<tr>
<td>$\sigma^2_{out,i}$</td>
<td>variance of the Gaussian variable at degree $i$ variable node output</td>
</tr>
<tr>
<td>$T_i$</td>
<td>the $i$th Taylor series coefficient of $1 - \rho^{-1}(1 - x)$</td>
</tr>
<tr>
<td>$(\lambda, \rho)$</td>
<td>degree distribution pair</td>
</tr>
<tr>
<td>$[\ ]$</td>
<td>smallest closest integer</td>
</tr>
<tr>
<td>$[\ ]$</td>
<td>closest integer</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

Low-Density Parity-Check (LDPC) codes were introduced by Gallager in the early sixties [1][2]. They however did not receive enough attention, and were soon forgotten. A short while after the successful introduction of turbo-codes [3], they were remembered and rediscovered by researchers [4]. They have received much attention since then.

LDPC parity-check codes are linear block codes that are decoded iteratively using message passing algorithms. The most important property of such algorithms is that while they can asymptotically achieve capacity, their decoding complexity grows almost linearly with code’s block length as opposed to maximum likelihood decoding method whose complexity grows exponentially with block length [5].
A lot of research has been done on LDPC codes that can be categorized into two important categories: performance analysis and design. Due to the nature of the associated decoding algorithms and their close-to-capacity performance, the number of code words that has to be tested in Monte Carlo simulation is so large for low bit error rates making it too complex to implement. Density Evolution (DE) was presented in [6][7] to overcome this problem in which we trace the evolution of the density of the messages passed within the decoder throughout iterations. It also gives the asymptotic performance limit of iterative decoding algorithms. To make the analysis even simpler, other methods such as Extrinsic Information Transfer (EXIT) chart [8][9] analysis have been proposed which are based on approximating the message densities with Gaussian distribution. Density Evolution and EXIT charts are also important tools to design LDPC codes.

This doctoral thesis is concerned with both of the above mentioned categories. As will be shown in the next chapter, some message passing algorithms; in particular the belief propagation (BP) algorithm, require knowledge of the channel noise power to perform optimally. We analyze the performance of the BP algorithm in the presence of channel estimation error, also referred to as mismatch, for different types of codes. We show that irregular codes that are optimized with the assumption of perfect channel estimation may not be good candidates in the presence of mismatch. We then focus on the more challenging problem of code design. We show that LDPC codes can be designed that perform well both in the presence and absence of mismatch. We propose a method to generate the EXIT charts in the presence of channel estimation error that are used for code design.
Although the EXIT chart method considerably reduces the complexity of code design compared to density evolution, it is still categorized as an optimization problem. These problems are usually time-consuming and do not provide any insight into the convergence properties of the ensembles and their relationship with the degree distributions. This encourages us to investigate whether there is a systematic approach to design good codes without optimization. We start by working on a simple channel model called the Binary Erasure Channel and show that for this channel, it is possible to design codes with close-to-optimum performance deterministically. We then extend this method to Binary Input Additive White Gaussian Noise (BIAWGN) channel. We propose a deterministic approach to design codes with close-to-optimum performance in both the presence and absence of mismatch. We show that the performance of the codes constructed based on our approach are very close to those already optimized based on density evolution and EXIT charts.

The organization of the thesis is as follows. In the next chapter we will have a brief review on communication channels and block codes. We then introduce LDPC codes and two important message passing algorithms to decode them when used over BIAWGN channels: Belief Propagation [6][7] and Min-Sum [10]. We then introduce density evolution technique as the main and the most accurate method to analyze the asymptotic performance of LDPC codes. Later in this chapter, we present some approximations that have been used to reduce the complexity of density evolution; in specific we review the EXIT chart method. These introductory parts prepare the reader to move on to the next chapters where we use the equations in Chapter 2 frequently.
Chapter 3 deals with the performance of the belief propagation algorithm in the presence of channel estimation error. In Chapter 2 we have already demonstrated that BP algorithm needs a correct estimate of the channel noise power to achieve its ultimate performance. We show that the performance degradation due to channel estimation error is highly dependent on the ensemble degree distributions. In this chapter, a method based on Gaussian approximation is developed to reduce the complexity of computing the thresholds compared to the case when one uses density evolution. We also investigate the performance of BP at two extreme cases: when we estimate the channel noise power much smaller and much larger than its actual value. The results of this chapter have been presented in [11] and published in [12] and [13].

In Chapter 4 we design the LDPC code ensembles in the presence of mismatch. We expect these codes to perform well for the case with perfect channel estimation and at the same time be robust against channel estimation error. It is seen that by a small sacrifice in the performance when no mismatch is present; it is possible to design irregular codes that perform well in the presence of it. We show how to generate the EXIT curves in the presence of mismatch and use them in the design process. The results of this chapter have been presented in [14], [15], [16] and submitted for publication in [17].

In Chapter 5, for the binary erasure channel we propose a method to deterministically design ensembles with close-to-optimum performance. We first consider a case where for a fixed channel parameter; we design codes with maximum possible rate. The results will then be extended to a more practical case where for a fixed rate, we design codes that provide reliable transmission for the worst possible channel
parameter. We show that using this method, codes with close-to-optimum performance can be designed. The results of this chapter will be presented in [18] and submitted to [19].

In Chapter 6 we show that using the method of Chapter 5 for the binary erasure channel, codes with close-to-optimum performance can be designed whose performance can achieve the channel capacity when the maximum variable node and check node degrees tend to infinity. We then prove some important properties of our proposed ensembles. We would like to note that Shokrollahi et. al. [20][21][22] have also designed a different class of capacity-achieving sequences. In this chapter, we will compare our approach with that of Shokrollahi and discuss its advantages. The results of this chapter will be submitted for possible publication in [23].

In Chapter 7 we extend the method proposed in Chapter 5 for deterministic design of LDPC codes over Binary Erasure Channel, to BIAWGN channel. We show that when no mismatch is present, LDPC codes can be designed whose performance are very close to those designed based on density evolution. We then consider a case similar to Chapter 4 where we want to design codes in presence of mismatch. We show that using an approach similar to the no-mismatch case, we can systematically design LDPC codes that perform well in presence of mismatch.

In Chapter 8 concluding remarks and possible future research will be presented.
Chapter 2 Background review of Low Density Parity Check codes

2.1 Digital communication systems

Assume that we want to transmit the data from a source to a destination. We can simplify the actual path in which the data is to be transmitted by a communication channel model. We propose a more clear definition below:

*Definition 2-1:*
A communication channel is defined to be a system consisting of a (continuous or discrete) random variable $X$ as its input, a (continuous or discrete) output random variable $Y$, and a mapping $\omega$ that relates $X$ to $Y$.

If the alphabet at the input of the channel is binary, we say the channel is *binary-input*.

We now review some important binary-input channel models:

**Binary Erasure Channel (BEC)**

This simple discrete channel is specified by its erasure probability $\varepsilon$. For this channel, the input alphabet is binary (0 and 1) and the output alphabet is ternary: 0, 1 and E. With probability $\varepsilon$, 1 and 0 are mapped to E, and with probability $1-\varepsilon$, 0 is mapped to 0 and 1 is mapped to 1 [24].

**Binary Symmetric Channel (BSC)**

This discrete channel is specified by its crossover probability $p$. For this channel, the input and output alphabets are binary (0 and 1). With probability $p$, 1 and 0 are mapped to 0 and 1 respectively, and with probability $1-p$, 0 is mapped to 0 and 1 is mapped to 1.

**Binary Input Additive White Gaussian Noise (BIAWGN) channel**

In this channel, 1 and -1 symbols are sent and a white Gaussian noise with zero mean and variance of $\sigma^2$ is added to the transmitted symbol. Therefore, the channel output $Y$ is a continuous random variable whose density is:

$$f_y(y) = \frac{1}{2\sqrt{2\pi}\sigma^2} (e^{(y-1)/2\sigma^2} + e^{(y+1)/2\sigma^2}).$$
Definition 2-2:
A binary input channel is called output-symmetric if [7]:
\[ f_y(y | 1) = f_y(-y | 1). \]
It is easy to see that BEC, BSC and AWGN channels are therefore output-symmetric.

Capacity
The capacity of BEC and AWGN channel can be computed as [24]:
\[ C(\varepsilon) = 1 - \varepsilon, \] (2-1)
and
\[ C(\sigma) = 1 - \int_{-\infty}^{\infty} e^{-\frac{(\beta-1)^2}{2\sigma^2}} \log_2 \left[ 1 + e^{-\beta} \right] d\beta, \] (2-2)
respectively.

2.2 Channel Coding

The idea in channel coding theory is to add redundancy to the information bits in such a way that they can be recovered in the receiver if they are corrupted by noise.

2.2.1 Linear block codes

Definition 2-3:
An \((n,k)\) linear block code over binary vector space \(F^n\) is a \(k\)-dimensional closed subspace \(C\) of \(F^n\).

Definition 2-4:
A generator matrix for an \((n,k)\) linear block code \(C\) is a \(k \times n\) matrix \(G\) whose rows form a basis for \(C\).

Definition 2-5:

A parity-check matrix for an \((n,k)\) linear block code \(C\) is a \((n - k) \times n\) matrix \(H\) whose rows form a basis for the null-space of \(C\).

It can be shown that having either \(G\) or \(H\), would completely describe the block code[5]. In this thesis we are only interested in binary codes.

2.2.2 Maximum Likelihood (ML) sequence decoding

This method tries to minimize the probability of frame (word) error. In this method, we are looking for a valid code word that has the minimum Euclidian distance with the received code word. This method is in general too complex as the number of computations and comparisons grows exponentially with the block length [5]. For block codes over the BSC, once we use the algebraic structures\(^{1}\)[25], the received codeword can be ML decoded efficiently. For the case of Reed Solomon codes, we can also achieve the capacity of BSC. These methods however, can not be extended to BIAWGN case. One way to efficiently decode block codes over the BIAWGN channel, is to use Maximum A Posteriori (MAP) decoding that will be discussed in the next subsection.

2.2.3 Bit level Maximum A Posteriori (MAP) decoding

\(^{1}\) For example Bose, Ray-Chaudhuri, Hocquenghem (BCH) codes
In this decoding method, the probability of bit error is minimized. For each bit, we compute the probability that the transmitted bit is 0 given the received codeword. We then compute the probability that the transmitted bit is 1 given the received codeword. By comparing these two values, we make a decision on that bit. This algorithm can be efficiently implemented in the logarithm domain. We define Log-Likelihood Ratio (LLR) of a bit as:

$$\lambda = \log\left(\frac{P(x = 0)}{P(x = 1)}\right)$$

(2-3)

The sign of the LLR determines the transmitted bit and its magnitude shows the reliability of the decision made. For the BIAWGN channel with noise variance $\sigma^2$, if the channel output is represented by $r$, the channel LLR's of the received bits can be computed as:

$$\lambda_{ch} = \log\left(\frac{P(x = 0 | r)}{P(x = 1 | r)}\right) = \log\left(\frac{P(r | x = 0)P(x = 0)}{P(r | x = 1)P(x = 1)}\right) = \frac{2}{\sigma^2} r.$$  

(2-4)

MAP decoding for convolutional codes can be done using the Bahl-Cocke-Jelinek-Raviv (BCJR) [26] algorithm. For block codes, it can be achieved using the Belief Propagation (BP) algorithm. The details of the BP algorithm will be explained in the next section.

2.3 Low-Density Parity-Check Codes

2.3.1 LDPC code structure

LDPC codes are linear block codes, and therefore can be fully described by their parity-check matrices. What distinguishes an LDPC code from an arbitrary block code is that its associated $H$ matrix is sparse meaning that a low percentage of the matrix elements are non-zero.
**Definition 2-6:**

A regular \((j, k)\) LDPC is a code whose parity check matrix is an \(M \times N\) binary matrix having exactly \(j\) ones in each column and exactly \(k\) ones in each row, where \(j < k\) and both are small compared to \(N\).

In a \((j,k)\) LDPC code, each bit is involved in \(j\) parity check equations and each parity check equation has \(k\) variables. There are total of \(N\) code bits and \(M\) parity check equations. If the \(H\) matrix is sparse but does not satisfy the above definition, its corresponding code is called an *irregular* LDPC code. For example, we may consider a parity check matrix for which half of the columns (respectively rows) have \(j_1\) (respectively \(k_1\)) ones and the other half have \(j_2\) (respectively \(k_2\)) ones.

**2.3.2 Tanner graph**

Any LDPC code may be represented by a *Tanner graph* [27]. A Tanner graph can be seen as a visual representation of the parity check matrix \(H\). Tanner graph contains \(N\) variable (bit) nodes, one for each bit, and \(M\) check nodes, one for each of the parity checks. The bit nodes are depicted using circles, while the check nodes are depicted using squares. The check nodes are connected to the bit nodes they check. Specifically, an edge connects check node \(m\) to bit node \(n\) if and only if the \(m\)-th parity check involves the \(n\)-th bit, or more mathematically, if and only if \(H(m,n) = 1\). The graph is said to be *bipartite* because there are two distinct types of nodes, bit nodes and check nodes, and there can be no direct connection between any two nodes of the same type.

*Example 2-1:* Consider the linear block code \(C\) specified by the parity-check matrix
2.3.3 Degree distributions

Based on Definition 2-6, there are more than one $H$ matrix which can describe a $(j,k)$-LDPC code. We say that the values of $j$ and $k$ define an LDPC code ensemble. It has been shown in [6][7] that for long block lengths, different random realizations of one ensemble perform very closely to each other. In an irregular LDPC code ensemble, only the percentage of columns (respectively rows) having $k$ (respectively $j$) ones is important. An LDPC code ensemble $(\lambda, \rho)$ is uniquely described by its variable node and check node degree distributions as:

$$\rho(x) = \sum_{i=2}^{D_c} \rho_i x^{i-1},$$  \hspace{1cm} (2-5)

and
\[ \hat{\lambda}(x) = \sum_{i=2}^{D_v} \hat{\lambda}_i x^{i-1}, \]  

(2-6)

with conditions:

\[ \sum_{i=2}^{D_v} \rho_i = 1, \]  

(2-7)

\[ \sum_{i=2}^{D_v} \lambda_i = 1, \]  

(2-8)

where the coefficient of \( x^i \) represents the fraction of edges connected to the nodes of degree \( i+1 \) and \( D_v \) and \( D_c \) represent the maximum variable node degree and maximum check node degree respectively.

Average variable node and check node degrees can be computed as:

\[ \bar{d}_v = \frac{1}{D_v} \sum_{i=2}^{D_v} \rho_i / i, \]  

(2-9)

\[ \bar{d}_c = \frac{1}{D_c} \sum_{i=2}^{D_c} \lambda_i / i, \]  

(2-10)

We can also show that if the matrix \( H \) has full rank, code rate \( R \) can be written as:

\[ R = 1 - \frac{\bar{d}_v}{\bar{d}_c}, \]  

(2-11)

2.3.4 Message passing decoders

The message passing decoding algorithms can be described using Tanner graphs. The decoding process is performed by exchanging messages between variable-nodes and check-nodes along the edges of the code’s Tanner graph, iteratively. We denote iteration number by \( \ell \). Before starting the algorithm \( (\ell = 0) \), variable-nodes and check-nodes are assigned some initial values based on the channel outputs. Then, every check-node
(variable-node) processes the messages it has received from its neighboring variable-nodes (check-nodes), along with its initial message, to produce new messages which it then sends back to its neighboring variable-nodes (check-nodes). A message-passing decoding algorithm performs well on a graph for which the outgoing message along an edge is independent of the incoming messages previously received along that edge. In other words, only extrinsic messages are passed along edges of the Tanner graph.

### 2.3.4.1 Belief Propagation (Sum Product) Algorithm:

The belief propagation algorithm can be considered as MAP decoding of a block code. As explained in Section 2.2.4, our aim is to minimize bit error rate and to achieve this goal, for each bit we have to compute its LLR based on the received noisy block. Applying this strategy, we will come up with the following algorithm (details can be seen in [28]). We first compute the channel LLR's of each bit \( \lambda_{ch} \) (for the case of BIAWGN channel we use (2-4)). These are the initial messages that are passed to the check nodes. At a check node of degree \( k \) and in the \( \ell \)th iteration, if the input messages are shown by \( \lambda_i \)'s, the following operation is preformed to produce the new message based on the extrinsic input messages:

\[
\lambda_{out} = 2 \tanh^{-1} \left( \prod_{i=1}^{k-1} \tanh \left( \frac{\lambda_i}{2} \right) \right). \tag{2-12}
\]

In a variable node of degree \( j \), the following operation is performed:

\[
\lambda_{out} = \sum_{i=1}^{j-1} \lambda_i + \lambda_{ch}. \tag{2-13}
\]

---

\[ ^2 \text{Such graph is called cycle free graph [27].} \]
When we reach the maximum number of iterations, the LLR on which we make the hard decision is computed as:

\[ \lambda_{\text{out}} = \sum_{i=1}^{l} \lambda_{i} + \lambda_{\text{ch}}. \]  

(2-14)

### 2.3.4.2 Min-Sum Algorithm

Equation (2-12) is not easy to implement. In fact, \( \text{tanh}' \) is not bounded when its input approaches 1 or -1. To avoid this and to reduce the computational complexity, a sub-optimum algorithm called Min-Sum (MS) has been proposed. The variable node operation of this algorithm is the same as BP while the check node operation is replaced by:

\[ \lambda_{\text{out}} = \text{sign}\left(\prod_{i=1}^{k-1} \lambda_{i}\right) \cdot \min_{\lambda_{i}} (|\lambda_{i}|). \]  

(2-15)

### 2.4 Asymptotic performance analysis of LDPC codes

#### 2.4.1 Density evolution

As discussed in Section 2.3.3, for large block lengths, different realizations of a code ensemble perform almost equally. This suggests that we might be able to predict the performance of a very long LDPC code (this is called the asymptotic performance) statistically based on its ensemble degree distributions, rather than simulation (e.g., the Monte Carlo method). In fact, we could follow the evolution of the density of the extrinsic messages exchanged between variable nodes and check nodes during iterations. This method is called density evolution (DE). The large block length also makes it possible to assume that for a given node, the neighboring graph is tree-like and cycle free[6][7][29]. This fact allows us to assume that the incoming messages to any node are...
independent. The independency assumption makes it possible to obtain the density of the output messages based on the densities of the input messages to variable nodes and check nodes. We now restate some definitions from [35][7].

**Definition 2-7:**
A degree $D_c$ check-node operator $\Psi$ is said to be symmetric if

$$\Psi(\lambda_1, \lambda_2, ..., \lambda_i, ..., \lambda_{D_c-1}) = -\Psi(\lambda_1, \lambda_2, ..., \lambda_{i-1}, \lambda_i, ..., \lambda_{D_c-1})$$

for $i \in \{1, 2, ..., D_c\}$

**Definition 2-8:**
A degree $D_v$ variable-node operator $\Psi$ is said to be symmetric if

$$\Psi(-\lambda_{ch}, -\lambda_1, -\lambda_2, ..., -\lambda_i, ..., -\lambda_{D_v-1}) = -\Psi(\lambda_{ch}, \lambda_1, \lambda_2, ..., \lambda_i, ..., \lambda_{D_v-1})$$

for $i \in \{1, 2, ..., D_v\}$

It can be shown that if the check node and variable node operations fulfill the conditions of Definitions 2-7 and 2-8 and if the decoder is used over an output-symmetric channel, the bit and frame error probabilities are independent of the transmitted code words. Throughout the thesis, we assume that the all-one codeword has been sent. This assumption is true because the channel is output symmetric and a linear block code is being considered.

Consider now a regular $(j,k)$ ensemble over the BIAWGN channel. The density of the initial LLR message $f_{v,t,i}$, is Gaussian with mean $2/\sigma^2$ and variance $4/\sigma^2$; $\mathcal{N}(2/\sigma^2, 4/\sigma^2)$ (see (2-4)). Assuming that all messages at the check node input are independent and identically distributed, we can compute the density of the check node output message based on (2-12). The resulting density $f_{c,t,k}$ (density at the output of the degree $k$ check node at $t$'th iteration) is now considered as the density at the variable node inputs. Based
on (2-13), to obtain the density at the output of the degree \( j \) variable node at \( \ell \)’th iteration \( f_{v,\ell,j} \), we have to compute the convolution of the variable node input densities. We continue this process and compute \( f_{v,\ell,j} \) and \( f_{c,\ell,k} \) iteratively until we reach the maximum number of iterations. We now use (2-14) to compute the density \( f_D \) on which we have to decide about the transmitted bit. The probability of error can be computed as:

\[
P_e = \int_{-\infty}^{0} f_D(x) dx . \tag{2-16}
\]

The indexes \( k \) and \( j \) in \( f \) can be dropped for regular codes.

For irregular codes, if the density that comes from variable nodes at iteration \( \ell \) is shown by \( f_{v,\ell} \), the density that has to be passed to variable nodes at the next iteration can be computed by linearly combining the computed densities for different check nodes of different degrees; \( f_{c,\ell,k} \)’s:

\[
f_{v,\ell} = \sum_{k=1}^{D_v} \rho_{\ell} f_{v,\ell,k} . \tag{2-17}
\]

This density is passed to variable nodes and the resulting density \( f_{v,\ell+1} \) can be computed as:

\[
f_{v,\ell+1} = \sum_{j=1}^{D_v} \lambda_{\ell} f_{v,\ell,j} . \tag{2-18}
\]

**Density evolution for BEC**

For discrete channels such as BEC and BSC, the densities depend only on one parameter (probability of message error in BSC and probability of message erasure for BEC). Therefore, DE can be carried out by tracking the evolution of a single parameter. At first iteration, the probability of a message erasure is equal to the channel erasure.
probability \( \varepsilon \). If we show the erasure probability of messages at iteration \( \ell \) by \( E(\ell) \), we have [20][21][22]:

\[
E(\ell) = \varepsilon \lambda (1 - \rho (1 - E(\ell - 1)))
\]  

(2-19)

where \( E(1) = \varepsilon \).

One can easily prove from (2-19) that if for \( 0 < x \leq 1 \),

\[
\varepsilon \lambda (1 - \rho (1 - x)) < x,
\]

(2-20)

the probability of erasure of the messages on each edge tend to zero as iterations tend to infinity. It is easy to see that (2-20) is equivalent to:

\[
\varepsilon \lambda (x) - 1 + \rho^{-1}(1-x) < 0, \quad 0 < x \leq 1.
\]

(2-21)

We call any code ensemble that satisfies (2-21) convergent for the given \( \varepsilon \).

2.4.2 Thresholds

It can be seen that the (2-19) and (2-20) depend on ensemble degree distributions and channel erasure probability. It can be proven that for a given code ensemble, there is a unique \( \varepsilon_0 \) for which if \( \varepsilon < \varepsilon_0 \), \( E(\ell) \) tends to zero as iterations continue and \( E(\ell) \) is bounded away from zero otherwise. This value is called the threshold of the code ensemble on BEC. For BIAWGN channel, the threshold is defined similarly.

Definition 2-9:

For a given ensemble, we define the SNR threshold as the infimum of all channel SNRs such that while performing density evolution, the bit error rate given by (2-16) converges to zero as the iteration number tends to infinity.

2.4.3 Symmetry (consistency) property

Definition 2-10:
A probability density function (PDF) $f(x)$ of a random variable $X$ is called symmetric (consistent) if and only if the following relationship holds:

$$f(x) = e^x f(-x). \quad (2-22)$$

It is proven in [6][7] that the PDF of an LLR conditioned on the transmitted bit is always consistent, in other words, a non-consistent function can not be the PDF of a conditional LLR. Another important property which was proven in [7] is that the BP operations in check node (2-12) and variable node and (2-13), preserve the consistency property. It means that if the input density to a check node (variable node) is consistent, so is the output. Consequently, while performing density evolution to analyze BP, all exchanged messages between variable nodes and check nodes are consistent. It can also be verified that the consistency of the PDF at the input of a check node (variable node) operation of BP, is necessary and sufficient condition for the output PDF to be consistent. As a result, if the input PDF to the nodes is not an LLR, the output can not be consistent. It can also be seen that the Min-Sum operation (2-15) does not preserve this property. Based on (2-22), a Gaussian random variable is consistent if and only if its variance is twice the absolute value of its mean. Throughout the thesis since we assume all-one codeword is transmitted, we may drop the term ‘absolute value’ for the mean.

2.5 Approximations to density evolution

2.5.1 Gaussian approximation

As discussed in the previous section, for BSC and BEC channels, density evolution consists of tracking a single parameter during iterations. For BLAWGN however, a continuous density has to be tracked. Based on the required accuracy in
computing the thresholds, we have to use different quantization levels. To get an accuracy of 0.1 dB, we usually need at least 6-bit quantization which means that we have to update a vector of length 64 at each iteration. To reduce the complexity of density evolution in BP algorithm, it was suggested in [29] to approximate the densities with Gaussian distribution. Since the densities are consistent, the Gaussian estimated version of the message is also consistent. It means that we only need to track one parameter which is usually the mean of the Gaussian variable. At the output of the variable nodes in BP, one sums up the channel LLR and the incoming extrinsic messages from the check nodes. Based on the central limit theorem, we expect that regardless of the distribution of the incoming messages, the output message has a distribution close to Gaussian. Using histogram estimation method based on Monte Carlo simulation, this assumption is confirmed. Although in the check node outputs, this assumption is not valid any more; Gaussian approximation method still would result in rather accurate threshold computations.

Consider a \((j,k)\) regular LDPC code. At variable nodes, the incoming extrinsic messages would be estimated by a consistent Gaussian variable with mean \(m_{in}\). Therefore, the variable node output would be a Gaussian random variable whose mean \(m_{out}\) is computed as:

\[
m_{out} = (j - 1)m_{in} + m_{ch},
\]

where \(m_{ch} = 4R \frac{E_b}{N_0}\) is the mean of the initial LLR values inputted to the belief propagation algorithm and \(E_b, N_0\) and \(R\) are the average energy per information bit, and the one-sided power spectral density and code rate, respectively.

---

3 A more detailed discussion on DE computational complexity can be found in Chapter 3.

4 It should be noted that a similar idea has been used for Turbo codes. The difference is that instead of mean, the error probability corresponding to a Gaussian variable is tracked.
We estimate the density of the random variable $\lambda_{\text{out}}$ at the output of the check node by a consistent Gaussian variable $\lambda_{\text{G,out}} \sim N(m_{\text{out}}, 2m_{\text{out}})$ that satisfies the following equation:

$$E\{\tanh(\lambda_{\text{out}}/2)\} = E\{\tanh(\lambda_{\text{G,out}}/2)\}, \quad (2.24)$$

where $E\{\cdot\}$ is the expectation operator. Equation (2-12) can be rewritten as:

$$\tanh(\lambda_{\text{out}}/2) = \prod_{i=1}^{k-1} \tanh(\lambda_i/2). \quad (2.25)$$

Therefore using the independency assumption of $\lambda_i$'s:

$$E\{\tanh(\lambda_{\text{out}}/2)\} = \prod_{i=1}^{k-1} E\{\tanh(\lambda_i/2)\}. \quad (2.26)$$

We rewrite (2-24) as:

$$E\{\tanh(\lambda_{\text{G,out}}/2)\} = \prod_{i=1}^{k-1} E\{\tanh(\lambda_i/2)\}. \quad (2.27)$$

We recall that the incoming extrinsic messages to check node are estimated by identically distributed Gaussian variables. Consequently if we define function $g$ by (2-28), (2-27) will be modified to (2-29) and (2-30):

$$g(x) = E(\tanh(M(x, 2x)/2)), \quad (2.28)$$

$$g(m_{\text{out}}) = [g(m_m)]^{k-1}, \quad (2.29)$$

$$m_{\text{out}} = g^{-1}([g(m_m)]^{k-1}). \quad (2.30)$$

We can summarize this methods into iteratively using (2-29) and (2-23) where we start by (2-29) and set $m_m = m_{\text{ch}}$. Function $g(.)$ can be pre-computed and stored in a look-up table.
The results reported in [29] state that for regular codes, this method can predict the thresholds that are at most 0.1 dB away from the actual thresholds computed by DE with high precision. For irregular codes, we use a linearly weighted summation of densities similar to DE method. More details can be found in [29] and in the next chapter.

2.5.2 EXIT charts

Extrinsic Information Transfer charts were used in [8][9] to analyze the asymptotic performance of LDPC codes. Before we proceed any further, we need to review some definitions. Consider a binary-input continuous-output channel whose input is \( X = \pm 1 \) and whose output \( Y \) has distribution \( f_{X}(y) = \frac{f_{X}(y | X = 1) + f_{X}(y | X = -1)}{2} \). The mutual information between \( X \) and \( Y \) can be written as:

\[
I(X;Y) = \frac{1}{2} \sum_{x=\pm1} \int_{-\infty}^{\infty} f_{X}(\xi | X = x) \log_2 \left( \frac{2f_{X}(\xi | X = x)}{f_{X}(\xi | X = -1) + f_{X}(\xi | X = 1)} \right) d\xi. \tag{2-31}
\]

If the channel is output-symmetric, (2-30) is modified to [30] :

\[
I(X;Y) = \int_{-\infty}^{\infty} f_{X}(\xi | X = 1) \log_2 \left( \frac{2f_{X}(\xi | X = 1)}{f_{X}(-\xi | X = 1) + f_{X}(\xi | X = 1)} \right) d\xi. \tag{2-32}
\]

If the density \( f \) is consistent, (2-31) can be written as:

\[
I(X;Y) = 1 - \int_{-\infty}^{\infty} f_{X}(\xi | X = 1) \log_2 (1 + e^{-\xi}) d\xi. \tag{2-33}
\]

If the density \( f \) is Gaussian with mean \( m \), function \( J \) defined in (2-33) would give the mutual information at the output of a BIAWGN channel\(^5\) on condition that the output is consistent:

\(^5\) Note that \( J(m) \) is the same as (2-2), the capacity of a BIAWGN channel if the Gaussian variable \( \mathcal{N}(m, \sigma^2) \) is scaled to \( \mathcal{N}(1, \sigma^2) \) where \( \sigma^2 = 2/m \).
The fundamental idea behind the EXIT chart is similar to what was discussed in Section 2.5.1. We would like to predict the performance of an iterative system by tracking only one parameter as opposed to DE. Similar to Section 2.5.1, we try to estimate densities by Gaussian random variables. The difference with Section 2.5.1 is the criterion based on which we choose the Gaussian variable. For the method of Section 2.5.1, the criterion was equation (2-24) at check node outputs. In the EXIT chart method, we estimate the check node output by a consistent Gaussian variable whose mean \( m_{out} \) satisfies the following equation:

\[
J(m_{out}) = I(X; \hat{A}_{out}) = 1 - \int_{-\infty}^{\infty} f_{\hat{A}_{out}}(\xi) \log_2 \left( 1 + e^{-\xi} \right) d\xi.
\] (2-35)

where \( f_{\hat{A}_{out}} \) is the density of the random variable at the check node output.

The operation at variable node remains the same as (2-22). As function \( J \) defines a one-to-one mapping between mean and mutual information, both parameters can be tracked throughout iterations. If we decide to track mutual information, we can use the following equation at variable nodes:

\[
J(m_{out}) = J((j-1)m_{in} + m_{ch}).
\] (2-36)

where \( m_{in} \) is the mean of a Gaussian variable whose mutual information \( I_A \) is equal to \( J(m_{in}) \). So we can write:

\[
I_E(I_A) = J(m_{out}) = J((j-1)J^{-1}(I_A) + m_{ch}).
\] (2-37)

Equation (2-36) is called the EXIT curve of the variable node which computes the mutual extrinsic information at the output of variable nodes as a function of mutual a priori
information at the variable node inputs. A similar curve can be derived for check nodes although there is no closed form of it\(^6\).

In general, to obtain the EXIT curve of an element such as a decoder in an iterative system we act as follows: Assume that the input mutual information is \(I_A\). We model the inputs to the decoder by consistent Gaussian variables whose mean is computed using (2-33) as \(m = \mathcal{J}'(I_A)\). Then, the PDF at the output of the decoder is obtained either analytically or by the histogram generated based on the Monte Carlo simulation. Finally using (2-32), the output mutual information is computed\(^7\). It is shown and verified in practice that mutual information is very robust against the changes in the shape of densities. It means that two different densities of the same mutual information at the input of a decoder would usually produce very close output mutual informations. In other words, the EXIT curve of a decoder can be assumed ‘almost’ independent from the decoder’s input and output densities. In Figure 2-2 we have shown the EXIT curves of a degree 3 variable node and a degree 6 check node which could predict the asymptotic performance of a regular (3,6) LDPC code. The iterations have been shown by trajectories between the two curves. We show the mutual information at the input and output of variable node and check nodes by \(I_{Ev}, I_{Av}\) and \(I_{Es}, I_{Ac}\) respectively. Another interesting property of mutual information can be seen when it is used to analyze irregular codes. Consider a degree distribution pair ensemble \((\lambda, \rho)\). Assume that we show the mutual information at the output of a degree \(j\) variable node and a degree \(k\)

\(^6\) EXIT curve of a decoder can be seen from a different point of view: \(I_A\) shows our primary information about the transmitted bit based on its received noisy version before passing it to the decoder. \(I_E\) shows the information about the transmitted bit after passing this data to the decoder. EXIT curve shows how the decoder improves our estimation about the transmitted bit.

\(^7\) For LDPC codes, the check node is in fact the decoder of a single parity check coder that computes the LLR of the parity of the bits on a parity check equation based on the LLR of each bit. Similarly, the variable node can be considered the decoder of a repetition code.
check node by $I^i_{Ev}$ and $I^k_{Ec}$, respectively. Then, the resulting overall mutual information at the output of variable nodes and check nodes can be computed by the linear combination of the constituent mutual informations and are given by (2-37) and (2-38) respectively [9]:

\[ I_{Ev} = \sum_{i=2}^{Dv} \lambda_i I^i_{Ev}(I_{Ac}, m_{ch}), \]

\[ (2-38) \]

\[ I_{Ec} = \sum_{i=3}^{Dc} \rho_i I^i_{Ec}(I_{Ac}). \]

\[ (2-39) \]

![Figure 2-2: Exchange of mutual information between variable nodes and check nodes in a (3,6) LDPC code ensemble at SNR=1.5dB.](image_url)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Chapter 3 Performance of LDPC codes in the presence of channel estimation error

In this chapter we first demonstrate the necessity of having a correct estimate of the channel noise power in BP algorithm. We then introduce a method based on Gaussian approximation that helps evaluate the threshold of an LDPC ensemble in the presence of channel estimation error. Using this tool, we investigate the effect of having an incorrect estimate of the channel noise power on the performance of regular and irregular LDPC code ensembles over the AWGN channel. This study is carried out asymptotically based
on the assumptions of a cycle-free graph and infinite block length. We also demonstrate that our asymptotic results are consistent with finite-length simulations.

3.1 Problem overview

In the belief propagation algorithm for LDPC codes over the binary input additive white Gaussian noise channel, the knowledge of the channel noise power is necessary to achieve the ultimate performance of the algorithm. Consider a BIAWGN channel with input $X$ that can take values $\pm 1$, and output $Y = X + N$, where $N$ is a Gaussian random variable with mean zero and variance $\sigma^2 \, (N \, (0, \sigma^2))$. In the log-likelihood ratio (LLR) domain, the input to the belief propagation algorithm for each bit should be $M_0 = 2Y/\sigma^2$. Starting from such initial messages, which is based on a perfect knowledge of the noise power $\sigma^2$, the belief propagation algorithm performs optimally in the sense that it provides the a-posteriori probabilities (APP) for bits assuming a tree-like (cycle-free) graph.

Now consider the case where the noise power is estimated erroneously as $\bar{\sigma}^2 \neq \sigma^2$. We refer to this as "signal to noise ratio (SNR) mismatch". The \textit{SNR mismatch ratio} $\eta$ is defined as $\eta = \sigma^2 / \bar{\sigma}^2$. The value of $\eta$ in dB is called "SNR offset". The performance of belief propagation with imperfect channel estimation has been addressed in [31][32][33] for turbo codes and in [34][35] for LDPC codes. In [34], simulation results for a few codes with finite block lengths are presented, while in [35], asymptotic analysis of belief propagation with mismatch in the context of magnetic recording channels for a regular LDPC code ensemble has been presented. For the AWGN channel, the results of [34][35]
indicate that the algorithm is more sensitive to the underestimation of SNR than it is to its overestimation.

In the presence of mismatch, we extend the definition of thresholds given in Section 2.4.2 for the no-mismatch case. For a given ensemble and an SNR offset \( \eta \), we define the SNR threshold \( a(\eta) \) as the infimum of all channel SNRs such that the bit error rate converges to zero as the iteration number tends to infinity. It is well-known that SNR thresholds in the presence of mismatch are always higher than that of no-mismatch case, i.e., \( a(\eta) > a(0) \), \( \forall \eta \neq 0 \) dB. We measure the sensitivity of an ensemble of codes to mismatch by the difference \( \Delta a(\eta) = a(\eta) - a(0) \) (dB).

To quantify the amount of tolerable mismatch for a given code ensemble at a given SNR \( s > a(0) \), we also define mismatch thresholds \( \beta^{(o)}(s) \) and \( \beta^{(u)}(s) \) for the over- and the underestimation of SNR, respectively. For over- (under)estimation, this is defined as the supremum (infimum) of SNR offset such that the bit error rate converges to zero as the iteration number tends to infinity. We use the difference \( \Delta \beta(s) = \beta^{(o)}(s) - \beta^{(u)}(s) \) as the measure of tolerance.

3.2 A simplified density evolution method based on Gaussian approximation

Although efficient ways have been proposed to implement the DE algorithm, it might be still too complex especially when one intends to compute thresholds for different code ensembles at different offset values. As mentioned in Section 2.5, alternative methods based on Gaussian approximation have been proposed. However, in the presence of
mismatch, these methods can not be applied directly. We remind that in the absence of mismatch, the input LLRs to belief propagation for all the bits are i.i.d. with distribution $:\mathcal{N}(2/\sigma^2, 4/\sigma^2)$. This distribution is symmetric (consistent) [6][7][29] with the ratio of variance to mean equal to two. As the symmetry property is preserved throughout the iterations of the belief propagation algorithm, the method of Gaussian approximation explained in Section 2.5.1, can be used to only track the mean of the Gaussian densities throughout the iterative process. This reduces the dimension of the problem from infinity (for full density evolution) to one, and significantly simplifies the analysis. For the case of mismatch, the inputs to belief propagation have distribution $:\mathcal{N}(2/\sigma^2, 4\sigma^2/\sigma^4)$. This not only violates the symmetry condition for the inputs to the decoder, but also causes the ratio of variance to mean to change with iterations (see Figure 3-1). Therefore, the Gaussian variables can not be tracked by only one variable and we need to track both the mean and the variance of them. For variable nodes, we still use (3-1) (or (2-23)) to track the means and employ the following equation (3-2) for variances:

$$m_{out} = (d_v - 1)m_{in} + m_0,$$

(3-1)

$$\sigma^2_{out} = (d_v - 1)\sigma^2_{in} + \sigma^2_0.$$  

(3-2)

For check nodes, we define the following two-input two-output mapping:

$$G(m, \sigma) = (g_1(m, \sigma), g_2(m, \sigma)) = (E[tanh(\mathcal{N}(m, \sigma^2)/2)], E[tanh^2(\mathcal{N}(m, \sigma^2)/2)])$$,  

(3-3)
where $E[.]$ denotes the expected value, and track the means and variances using

$$(m_{out}, \sigma_{out}) = G^{-1}(g_1^{d-1}(m_{in}, \sigma_{in}), g_2^{d-1}(m_{in}, \sigma_{in})).$$

(3-4)

Figure 3-1: Variance to mean ratio at the output of variable nodes as a function of iteration number for a (3,6)-regular ensemble with SNR offsets $\eta = -2, 0$ and $2$ dB.
We compute the mappings $G$ and $G^\prime$ numerically and use look-up tables to track the means and variances. In Figure 3-2, we have shown the functions $g_1(m,\sigma)$ and $g_2(m,\sigma)$, respectively.

To generalize this analysis to irregular codes similar steps as those described in Section 2.5.1 are taken. Assume an irregular code ensemble with degree distributions $\lambda(x)=\sum_{i=2}^{\infty} \lambda_i x^{-i}$ and $\rho(x)=\sum_{i=2}^{\infty} \rho_i x^{-i}$ for variable nodes and check nodes respectively. The distribution of messages passed between variable nodes and check nodes in this case are approximated by Gaussian mixtures. Suppose that, at a given iteration $\ell$, the Gaussian mixture at the output of check nodes and the input of variable nodes has the mean and the variance, $m_{in}$ and $\sigma^2_{in}$, respectively. For variable nodes of degree $i$, we estimate the output by a Gaussian random variable and compute its mean and variance using

$$m_{out,i} = (i-1)m_{in} + m_0,$$

$$\sigma^2_{out,i} = (i-1)\sigma^2_{in} + \sigma^2_0,$$

respectively. The density of the messages from variable nodes to check nodes is then approximated by the Gaussian mixture $\sum \lambda_i N(m_{out,i}, \sigma^2_{out,i})$. We now define $h_1$ and $h_2$ as follows:

$$(h_1, h_2) = \sum_{i=2}^{\infty} \lambda_i G(m_{out,i}, \sigma^2_{out,i}).$$

The output of a check node of degree $j$ is then approximated by a Gaussian random variable with the following mean and variance

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 3-2: Mapping $G$: (a) $g_1(m,\sigma)$, (b) $g_2(m,\sigma)$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The distribution of messages from check nodes to variable nodes is approximated by the Gaussian mixture \( \sum \rho_j N(m_{out,j}, \sigma_{out,j}^2) \). The mean and the variance of this distribution are obtained by the following equations to be used as inputs to variable nodes for iteration \( \ell + 1 \).

\[
(m_{out,j}, \sigma_{out,j}^2) = G^{-1}(h_{i}^{i+1}, h_{i}^{i+1}).
\]  

(3-8)

To examine the accuracy of the proposed method, we have compared its threshold results to those generated by density evolution for a few cases. Our results indicate that with steps of 0.01 for the mean and the variance, the difference between the exact and the approximate thresholds is at most about 0.1dB and 0.2dB for regular and irregular codes, respectively. We note that this inaccuracy is mainly due to the Gaussian approximation [29], but marginal improvement can be obtained by reducing the step size.

3.3 Complexity comparison with density evolution

Density evolution, as a tool for asymptotic analysis of iterative decoding, was first proposed in [6][7]. A more efficient implementation, referred to as discretized density evolution, was then presented in [36][37]. This implementation was further improved in

\[\text{In line with our results, it can be seen in [25] as well that the Gaussian approximation is generally less accurate for irregular codes.} \]
[38] in a number of ways, many of them exploiting the symmetry of the message distribution. In the case of mismatch, the symmetry condition is violated and thus we only compare our proposed method with that of [37]. We however acknowledge that using some of the modifications described in [38], the method of [36][37] can still be improved.

The density evolution of [37] is based on discretizing the messages to \( \Omega \) quantization levels and processing their corresponding \( \Omega \)-point probability mass functions (PMF) throughout iterations. To compare the complexity of the density evolution of [37] and that of our method, we consider a regular \((j, k)\) LDPC code.

At variable nodes, our method only requires two multiplications and two additions as described in (3-1) and (3-2). In comparison, for the density evolution of [37], the variable node operation consists of the fast Fourier transform (FFT) calculation of the PMF of the incoming messages from the check nodes, the \((j-2)\)-fold point-wise multiplication of this FFT, also multiplied by the FFT of the PMF of the initial messages, and finally an inverse FFT of the result.

At check nodes, our algorithm requires the storage of four 2-dimensional look-up tables for \( G \) and \( G^t \). The calculation of the output mean and variance then only involves four table look-ups and \( 2(k-2) \) multiplications as can be seen in (3-3) and (3-4). For the density evolution of [32], one has to store a 2-dimensional lookup table for the map of belief propagation check node updates. Using this table, the algorithm calculates a discrete convolution of the input PMFs, two at a time. This adds up to \((j-2)\) discrete convolutions each with \( \Omega \) points. The complexity of this calculation is quadratic in \( \Omega \).
Based on the above discussions, the complexity of discretized density evolution not only is a function of $j$ and $k$ but also depends strongly on $\Omega$. Our experiments show that for the same level of accuracy as that of our method, discretized density evolution requires $\Omega = 64$ to 256, corresponding to 6- to 8-bit quantization, depending on the degree distributions and offset values. For these values of $\Omega$, one can see that our method handily beats discretized density evolution in complexity.

We end this part by noting that the simplicity of our algorithm compared to density evolution is particularly helpful for cases where a large number of threshold calculations are required or for the design of irregular LDPC codes at the presence of mismatch.
3.4 Asymptotic performance of belief propagation at offset limits

SNR underestimation

Assume that the SNR offset tends to minus infinity. In this case the estimate $\tilde{\sigma}^2$ of the channel noise variance is very large. This makes the initial LLR values of $m_0 = 2y/\tilde{\sigma}^2$ very small. Consider the first iteration of belief propagation, where the initial messages are passed from variable nodes to check nodes. Without loss of generality, consider a check node of degree three with two non-negative extrinsic inputs $L_1$ and $L_2$. The output is then equal to (see, e.g., [39])

$$L_3 = \min(L_1, L_2) + \ln(1 + e^{-(L_1+L_2)}) - \ln(1 + e^{-|L_1-L_2|}), \quad (3-11)$$

Using the Taylor series, for small values of $x$, we have $\ln(1 + e^{-x}) \approx \ln(2) - x/2$. Applying this to the second and the third terms of (3-11), we obtain $L_3 \approx 0$. This implies that the messages passed by check nodes to variable nodes are negligible compared to the initial LLR values. This situation will continue throughout the iterations and thus the bit error rate tends to that of the channel, which is equal to $Q(1/\sigma)$, where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int e^{-x^2/2} dx.$$

SNR overestimation

In the case of SNR overestimation, the initial LLR values, and thus all the messages passed from variable nodes to check nodes throughout the iterations, are very large. In this case, as the magnitude of the difference between the second and the third terms of (3-11) is upper bounded by $\ln(2)$, we have $L_3 \approx \min(L_1, L_2)$. This is the check node
operation of min-sum (see, e.g.,[10] [39]). As the variable operations of belief propagation and min-sum are also the same, this implies that the performance of belief propagation tends to that of min-sum as the SNR offset tends to plus infinity.

3.5 Effect of the channel estimation error on the performance of LDPC codes

We consider different ensembles of regular LDPC codes and investigate dependency of the sensitivity of belief propagation to the SNR mismatch on the degrees of variable and check nodes. In Figure 3-3, we have plotted the normalized SNR thresholds $\Delta\alpha(\eta)$ for a number of $(j,k)$ ensembles and for the range of SNR offset $\eta$ between -6 and +6dB. As can be seen, the sensitivity can change significantly by the variation of $j$ and $k$, particularly for the SNR overestimation. For example, while 4dB of SNR overestimation degrades the SNR threshold by only about 0.1dB for $(3,27)$ codes, it degrades the threshold by more than 1.1dB for $(5,6)$ codes. In general, the sensitivity increases by increasing the variable node degree and by decreasing the check node degree. It also appears that the effect of variable node degree is more prominent than that of the check node degree such that for a given rate $R = 1-j/k$, the codes with smaller $j$ and $k$ are less sensitive to mismatch. For example among rate-$\frac{1}{2}$ ensembles, the $(3,6)$ codes are the least sensitive, followed by $(4,8)$ and $(5,10)$ ensembles, respectively. Figure 3-3 also shows that while in general the SNR underestimation is more deteriorating than the SNR overestimation, for some ensembles and over a certain range of SNR mismatch, overestimation is almost as harmful as underestimation is (see, e.g., $(5,6)$ and $(6,8)$ ensembles for $\eta \in [-2,2]$).
Curves of Figure 3-3 asymptotically tend to the performance gap between belief propagation and min-sum as $\eta \to +\infty$. This implies that this gap decreases by decreasing $d_v$ and increasing $d_c$ with the effect of $d_v$ being more profound. As an example, for rate-$\frac{1}{2}$ regular LDPC codes, the performance gap between belief propagation and min-sum is about 0.6 dB for (3,6) ensemble [10]. This increases to about 0.88 dB and 1.06 dB for (4,8) and (5,10) ensembles, respectively [10].

![Figure 3-3: Normalized threshold values $\Delta \alpha(\eta)$ for a number of $(d_v,d_c)$ ensembles of regular LDPC codes and for the range of SNR offset $\eta$ between -6 and +6 dB.](image)
To verify the asymptotic results, we consider the ensemble of (3,6)-regular LDPC code. In Figure 3-4, we have plotted BER vs. SNR offset for a randomly constructed (3,6)-regular code of length $10^5$. As can be seen, the waterfall region of the BER curves starts at close vicinity of the computed threshold values. (For all the BER simulations in this chapter, the maximum number of iterations is 100 and at least 50 codeword errors have been observed.) We have also computed the mismatch thresholds $\beta^{(s)}(s)$ and $\beta^{(s)}(5)$ for $s = 1.5$, 2.0 and 2.5 dB, using our proposed Gaussian approximation method. These are shown in Figure 3-4. Note that $\beta^{(s)}(2) = \beta^{(s)}(2.5) = +\infty$. This is in agreement with the fact that both SNR values are larger than the min-sum threshold for this ensemble which is about 1.7 dB.

For the irregular codes, we consider 3 ensembles. The degree distributions for variable and check nodes of the first ensemble are

$$\lambda_1(x) = 0.24x + 0.21x^2 + 0.04x^3 + 0.12x^4 + 0.39x^5,$$

$$\rho_1(x) = x^7,$$

respectively. This ensemble, which is check-regular, is a slight modification of the first ensemble in Table II of [7]. We consider 2 more rate ½ irregular ensembles described by the following degree distributions [7], Table I:

$$\lambda_2(x) = 0.38354x + 0.04237x^2 + 0.57409x^3,$$

$$\rho_2(x) = 0.24123x^4 + 0.75877x^5,$$

and [7], Table II:

$$\lambda_3(x) = 0.19606x + 0.24039x^2 + 0.00228x^3 + 0.00221x^7 + 0.04088x^8 + 0.00150x^9,$$

$$\rho_3(x) = 0.00749x^7 + 0.99101x^8 + 0.00150x^9$$
Figure 3-4: BER of a randomly constructed (3,6)-regular LDPC code of length $10^5$ at different SNR values in the presence of SNR mismatch. Corresponding mismatch thresholds are also given.

Figure 3-5 shows that while the ensemble with $D_v = 30$ outperforms the other two ensembles in the absence of mismatch, the situation changes in the presence of mismatch. The ensemble with $D_v = 15$ takes over outside the range of $\eta \in [-0.7\,\text{dB}, 1.5\,\text{dB}]$, while beyond the SNR mismatch of about $+4.9\,\text{dB}$, the ensemble with $D_v = 4$ is superior.

In Figure 3-6 we have considered the first irregular ensemble and plotted BER vs. SNR offset for a randomly constructed code of length 20000. As can be seen, the waterfall region of the BER curves starts at close vicinity of the computed threshold values.
Figure 3-5: SNR threshold values for three rate 1/2 ensembles of irregular LDPC codes for the range of SNR offset $\eta$ between -6 and +6 dB.

Figure 3-6: BER of a randomly constructed rate-1/2 irregular LDPC code of length 20,000 with degree distributions $(\lambda, \rho)$. Corresponding mismatch thresholds are also given.
3.6 Concluding remarks

In this chapter we investigated the effect of mismatch on the performance of BP for different degree distributions. For a given code rate, we generally are looking for a code with the smallest possible SNR threshold and largest robustness to mismatch. For example, among rate-1/2 codes, the (3, 6) ensemble is the least sensitive one to mismatch. It is also known that for zero mismatch, this code ensemble has the best performance in terms of SNR threshold [10]. So, the (3,6) ensemble has the best performance in both the presence and the absence of mismatch. For irregular codes however, an ensemble that performs very well in the absence of mismatch may have a considerably worse performance in the presence of it. This motivates us to look for irregular ensembles that while providing low SNR thresholds in the absence of mismatch, are also robust against mismatch. This will be the subject of the next chapter.
Chapter 4 Design of LDPC codes in the presence of channel estimation error

4.1 Motivation

In this chapter we consider designing irregular LDPC codes in the presence of mismatch. As a motivation to this chapter, we remind from Section 3.6 that in general for a given code rate, one is usually interested in having a code with the smallest possible SNR threshold and largest robustness to mismatch. For irregular codes as opposed to regular codes, an ensemble that performs very well in the absence of mismatch may have a considerably worse performance in the presence of it. In Figure 4-1 we have shown the
SNR thresholds at different offset values of a rate-1/2 highly irregular code ensemble $C$ with the maximum variable node degree 20 by starred solid line. This ensemble has been optimized for BP in the absence of mismatch [36]. As can be seen, the ensemble is highly sensitive to SNR mismatch. In particular, SNR offsets of +3dB and -3dB increase the threshold by about 0.7 dB and 1.2 dB, respectively. We have also shown in Figure 4-1, by circed solid line, the BP performance of another rate-1/2 irregular code ensemble $C'$, again with maximum variable node degree 20. This ensemble is optimized for min-sum algorithm [36]. Figure 4-1 shows that although $C'$ performs worse than $C$ at offset zero, it is much less sensitive to SNR mismatch than $C$ is. In particular, for SNR offsets of +3dB and -3dB, the threshold increases by only about 0.1 dB and 0.6 dB, respectively. Although when no mismatch is present, $C$ outperforms $C'$ by about 0.4 dB, this superiority in performance lasts only for offset values in the range [-2dB, 2dB]. This example suggests that it is possible to reduce the sensitivity of BP with respect to SNR mismatch by sacrificing the performance at zero mismatch.

It is well-known that MS algorithm is insensitive to mismatch. One may therefore suggest using MS instead of BP to combat the deteriorating effect of mismatch. Although MS is fully insensitive to mismatch, for highly irregular codes, it can experience more than 1dB performance loss compared to BP at offset zero. For example, Figure 4-1 shows a performance degradation of about 1.2 dB for ensemble $C$. Although for irregular codes optimized for MS, the degradation in performance of MS compared to that of BP at offset zero is much smaller (e.g., for $C'$, this is about 0.2 dB), it is still non-negligible. This, added to the performance loss due to the sub-optimality of the ensemble for BP, results in considerable performance degradation at offset zero. For $C'$, as shown in Figure 4-1, the
total degradation of MS compared to the BP performance of $C$ at offset zero is about 0.6 dB. The above discussions motivate us to design irregular codes that are robust against mismatch and yet perform well in the absence of mismatch, when decoded by BP.

In this chapter, we show, both asymptotically and by finite-length simulations, that irregular codes can be designed that not only perform very well in the absence of mismatch, but also are more robust against mismatch compared to codes of equal rate and equal maximum variable node degree, optimized for the no-mismatch case.

![Figure 4-1: BP (---) and MS (- - -) thresholds for 2 irregular codes with maximum variable node degree of 20.](image)
4.2 Problem formulation

Consider an SNR offset range of interest $[\eta_1, \eta_2]$, where $\eta_1$ and $\eta_2$ are real numbers, usually negative and positive respectively. In this chapter, we are interested in designing an irregular code ensemble of certain rate for which the largest SNR threshold over the offset range $[\eta_1, \eta_2]$ is minimized over all possible degree distributions. This is a min-max optimization problem. In most cases, the search over the degree distributions is constrained by the maximum variable and check degrees or by certain constraints on the degrees of the constituent variable and check nodes.

It is well-known that SNR offset degrades the performance of BP, and that the larger the offset the worse the performance. Therefore, if $0 < \eta_1 < \eta_2$ ($\eta_1 < \eta_2 < 0$), the problem is reduced to minimizing $\alpha(\eta_2)$ ($\alpha(\eta_1)$). It is also well-known that underestimation of SNR is more deteriorating than overestimation [31]-[35]. In the case that $\eta_1 < 0 < \eta_2$ and $\eta_1 \leq \eta_2$, the optimization problem is then reduced to minimizing $\alpha(\eta_1)$. This includes the special case of $\eta_1 = -\eta_2$, which may be of more practical interest. Finally, for the case where $\eta_1 < 0 < \eta_2$ and $\eta_1 \geq -\eta_2$, the problem reduces to minimizing the maximum of $\alpha(\eta_1)$ and $\alpha(\eta_2)$.

4.3 EXIT charts in the presence of mismatch

Different methods have been proposed for designing irregular LDPC codes. One well-known method is density evolution [35][6][7]. For codes on BIAWGN channel however, this method has high complexity. Other methods, such as those in [8][9][29], have been proposed to reduce the complexity. The EXIT chart method proposed in [8][9] is based on the assumption that there is no SNR mismatch and thus the initial messages at
the input of the decoding algorithm are consistent. Moreover, the messages maintain this
case throughout the iterations. As explained in Section 2.5.2, to generate the EXIT
charts in the absence of mismatch, the input a priori information to check nodes (variable
nodes) is modeled as a consistent Gaussian variable with a given mutual information and
the resulting mutual information at the output of check nodes (variable nodes) is
computed. In the mismatch case, as the variance-to-mean ratio of the initial Gaussian
message is no longer 2, the consistency condition is violated. Therefore, to generate
EXIT curves, we can no longer model the input of check nodes (variable nodes) by a
consistent Gaussian variable and therefore EXIT charts can not be generated as explained
in [8][9]. For the case of mismatch, similar to the no-mismatch case, the EXIT chart
analysis is still based on tracking the mutual information between the transmitted bit and
the messages exchanged between the variable nodes and the check nodes. Suppose that
we would like to generate the EXIT curve of variable nodes of degree $j$ (check nodes of
degree $k$) for a given offset $\eta$ and a given SNR $\gamma$. For this, we select a check node degree $j$
(a variable node degree $k$) small enough (large enough) such that $\gamma$ is above the threshold
$\alpha(\eta)$ of the regular $(j,k)$ ensemble. We then consider this ensemble and assume that the
all-one code word is transmitted. Using density evolution, we can obtain the densities of
the messages at the input and the output of both variable nodes and check nodes at each
iteration. We then use (2-31) to compute the corresponding mutual information values.
To derive the EXIT curves from the $(I_a, I_E)$ points, we use polynomial curve fitting with
maximum degree 6. The EXIT charts for irregular codes are still obtained using (2-37)
and (2-38)\(^9\).

\(^9\) One should note that although we compute the EXIT charts using density evolution, this is performed
only once to generate the charts for the constituent variable and check nodes. The proposed method then

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In Figure 4-2, we have shown the EXIT charts for different offset values for a degree-3 variable node and a degree-6 check node at SNR = 4.2dB. As can be seen in Figure 4-2, regardless of the value of SNR offset, the boundaries of all the curves for variable nodes are \((I_{v}, I_{\theta}) = (0, J(4RE_{b}/N_{0}))\) and \((1,1)\), while these boundaries for all the check node curves are \((0,0)\) and \((1,1)\). Figure 4-2 also shows that any value of mismatch degrades the performance as it makes the tunnel between the check node and the variable node curves narrower. We were able to analytically prove this property for variable nodes. Before we state the theorem, we will go through some reminders and lemmas.

**Reminder 4-1:** If \(X\) and \(Y\) are two random variables whose mutual information is given by \(I(X;Y)\), multiplication of any of them by a constant does not change the mutual information. In other words:

\[
I(X, Y) = I(aX, bY), a, b \in \mathbb{R}
\]

**Proposition 4-1:** Assume that we have a Gaussian random variable \(X\) conditioned on binary random variable \(B = \{b| b=-1,1\}\) where \(X \sim \mathcal{N}(bm, \sigma^{2})\). Assume that mutual information between \(X\) and \(B\) is given by \(I(B,X)\). If we want to find a consistent random variable with the same mutual information, we have to multiply \(X\) by \(2m/\sigma^{2}\).

**Proof:** The resulting variable would have a normal distribution of \(\mathcal{N}(b2m^{2}/\sigma^{2}, 4m^{2}/\sigma^{2})\), which is a consistent Gaussian variable and based on Reminder 4-1, has the same mutual information. ■

uses linear combinations of these curves, as shown in (2-38) and (2-39) in a curve-fitting process to design irregular codes (the process will be explained in the next section). This is much less complex than directly using density evolution in the design process, where one would need to apply density evolution to any tested ensemble of irregular codes in the search process.
Figure 4-2: Degree-3 variable node and degree-6 check node EXIT charts for different offset values at SNR= 4.2dB.

**Proposition 4-2:** Assume that we have a consistent Gaussian random variable conditioned on binary random variable $B = \{b | b=-1,1\}$ where $X \sim \mathcal{N}(bm,2m)$. Assume that mutual information between $X$ and $B$ is given by $I(B,X)$. If we want to find a Gaussian random variable with the same mutual information for which the ratio of variance to absolute value of the mean is equal to constant $a$, we have to multiply this variable by $a/2$.

**Proof:** The resulting variable would have a normal distribution of $\mathcal{N}(bm/2, a^2m/2)$ and it can be verified that the variance to mean ratio is equal to $a$. ■

**Reminder 4-2:** Function $J$ defined in (2-33), accepts the mean of a consistent Gaussian random variable and gives the corresponding mutual information. Also, taking $J'$ of a given mutual information would generate the mean of a consistent Gaussian random variable that corresponds to the given mutual information.

In what follows, we assume that all-one code word is transmitted.
Theorem 4-1: Assume that $I_{E_1} = g(I_A)$ and $I_{E_2} = g(I_A)$ are the EXIT curves of a variable node of degree $j$ for offset $= 0$ and offset $\neq 0$ cases respectively. For any $I_A$ we have $I_{E_1} \geq I_{E_2}$.

Proof: Let the distribution of the received message form the channel (which we have erroneously assumed to be the true channel message LLR) be $\mathcal{N}(m_{ch}, \sigma^2_{ch})$ for which $\sigma^2_{ch}/m_{ch} = \alpha_{ch}$. Assume that we show the Gaussian estimated message at the output of check node which would be the input to the variable node by $\mathcal{N}(m_{in}, \sigma^2_{in})$. Also assume that the variance to mean ratio for this variable is $\alpha$. Using Proposition 4-2, we have:

$$m_{in} = \alpha J^2(I_A)/2, \quad (4-1)$$

$$\sigma^2_{in} = \alpha^2 J^2(I_A)/2, \quad (4-2)$$

We show $J^2(I_A)/2$ by $A$ for simplicity. Now assume that we represent the variable node output by $\mathcal{N}(m_{out}, \sigma^2_{out})$. We can write:

$$m_{out} = m_{ch} + (j-1)m_{in}, \quad (4-3)$$

$$\sigma^2_{out} = \sigma^2_{ch} + (j-1)\sigma^2_{in} = \alpha_{ch} m_{ch} + (j-1)\sigma^2_{in}. \quad (4-4)$$

To be able to compute the corresponding mutual information using $J(.)$, we use Proposition 4-1 and write:

$$m_{eq} = \frac{2m_{out}}{\sigma^2_{out}} m_{out} = \frac{2(m_{ch} + (j-1)\alpha A)^2}{\alpha_{ch} m_{ch} + (j-1)\alpha^2 A} = f(\alpha), \quad (4-5)$$

where $m_{eq}$ is the mean of the consistent Gaussian random variable that has the same mutual information as the variable node output. Now we compute:
\[ I_{E2}(\alpha) = J(m_{eq}) \]  

(4-6)

It can be shown by differentiation that the function \( f \) is maximum when \( \alpha = \alpha_{ch} \). In other words, for any given \( I_A \) and \( \alpha \), the resulting \( m_{eq} \) is always equal or smaller than \( m_{eq}^{\alpha=\alpha_{ch}} \).

Knowing that function \( J \) is an increasing function of \( m \), we can conclude that \( I_{E2}(\alpha) \leq I_{E2}(\alpha_{ch}) \) for any \( \alpha \). Now we obtain the value of \( m_{eq} \) at \( \alpha = \alpha_{ch} \):

\[
    f(\alpha_{ch}) = \frac{2(m_{ch} + (j-1)\alpha_{ch} \bar{A})}{\alpha_{ch}} = \frac{2m_{ch}}{\alpha_{ch}} + (j-1)J^{-1}(I_A),
\]

(4-7)

So:

\[
    I_{E2}(\alpha_{ch}) = J\left(\frac{2m_{ch}}{\alpha_{ch}} + (j-1)J^{-1}(I_A)\right),
\]

(4-8)

Based on Proposition 4-2, \( 2m_{ch}/\alpha_{ch} \) is equal to the mean of a consistent Gaussian variable with the same mutual information. We show this mean by \( m_{ch-no-mismatch} \) and write:

\[
    I_{E2}(\alpha_{ch}) = J(m_{ch-no-mismatch} + (j-1)J^{-1}(I_A)).
\]

(4-9)

This is the same as (2-36); the variable node operation if no mismatch was present. Since this holds for any \( I_A \), we conclude that

\[
    I_{E2}(\alpha_{ch}) = I_{E1}.
\]

(4-10)

Using (4-5) and (4-9) we conclude that:

\[
    I_{E2} \leq I_{E1}
\]

This proves the theorem.

Corollary 4-1: Proof of the theorem shows that if the variance to mean ratio of channel LLR is \( \alpha_{ch} \), we cannot model check node outputs with Gaussian variables with \( \alpha = \alpha_{ch} \); as in this case, the variable node in the presence of mismatch would be the same as that of no-mismatch case which is not true.
Corollary 4-2: For any given offset, the variable node EXIT curves starts at the same point as the variable node curve of no-mismatch case and ends at $I_{E2} = 1$.

Proof: If $I_A = 0$, then $A = 0$ and based on (4-4) we have:

\[
m_{eq} = 2m_{ch}/a_{ch} = m_{ch-no-mismatch}
\]

\[
I_{E2}(\alpha = a_{ch}, I_A = 0) = J(m_{eq}) = J(m_{ch-no-mismatch}) = I_{E2}(I_A = 0)
\]

which proves the first part. For $I_A = 1$, $A$ tends to infinity which causes $m_{eq}$ to tend to infinity too and we have:

\[
I_{E2} = J(m_{eq}) = 1. \blacksquare
\]

We end this sub-section by reminding that the curves in Figure 4-2 clearly demonstrate that the algorithm is more sensitive to the underestimation of SNR than it is to SNR overestimation. To verify the accuracy of the generated EXIT charts, we have used them to compute the thresholds of a few regular and irregular code ensembles. The results are at most 0.1 dB away from the results computed by density evolution.

4.4 Some examples of code design in the presence of mismatch

For irregular codes, variable node and check node degree distributions must satisfy (2-7), (2-8) and (4-10) (similar to (2-11)) conditions:

\[
R = 1 - \frac{\int \rho(x)dx}{\int \lambda(x)dx}
\]

which we have assumed the true rate to be equal to the design rate, i.e., there are no redundant check equations. In general, the design of irregular codes using EXIT charts is based on a curve-fitting procedure involving the variable nodes and check nodes EXIT curves, (see, e.g., [8]). To make the procedure simpler, it is often assumed that the
ensemble is check-regular. In this thesis, as the designed codes are compared to those of [8][36], we allow both the variable nodes and the check nodes to be irregular. Similar to [8][36], we choose two consecutive nonzero check degrees. Degree distributions are found that satisfy (2-7), (2-8) and (4-11) for a given rate $R$ and that require the minimum value of $E_b/N_0$ to leave the tunnel between the check nodes and the variables nodes curves open.

As a point of reference for our designed codes, we consider a code ensemble designed in [36]. This ensemble, referred to as $C_I$ in the rest of the thesis, is optimized for BP, and has rate $\frac{1}{2}$ and maximum variable node degree 20. The degree distributions for $C_I$ are:

$$
\lambda_1(x) = 0.234029x + 0.212425x^2 + 0.146898x^5 + 0.102840x^6 + 0.303808x^{19},
$$

$$
\rho_1(x) = 0.7187x^7 + 0.2813x^8.
$$

This ensemble is in fact the same as the one denoted by $C$ in Section 4.1. The starred solid line in Figure 4-3 shows the threshold values of this ensemble in the presence of mismatch for $\eta$ ranging from -6 to 6dB.

Suppose that the SNR offset range of interest is $[\eta_1, \eta_2] = [-2dB, 2dB]$. Based on the discussions of Section 4.2, in this case, our optimization problem is reduced to minimizing the threshold at $\eta = -2dB$. By constraining ourselves to a similar set of constituent variable degrees as in [8][36], we obtain the following degree distributions ($C_2$):

$$
\lambda_2(x) = 0.258x + 0.251x^2 + 0.269x^6 + 0.222x^{19},
$$

$$
\rho_2(x) = 0.342x^6 + 0.658x^7.
$$

The thresholds for this ensemble are shown in Figure 4-3 with squared solid line. As can be seen, in the offset range of [-2dB, 2dB], the maximum threshold has decreased by
about 0.15 dB compared to $C_j$. This is while the threshold at $\eta = 2$ dB has decreased even more than 0.2 dB, and the designed ensemble outperforms $C_j$ for any offset value outside the interval [-1 dB, 1 dB]. This is at the expense of a slight degradation of performance (less than 0.1 dB) at offset zero.

The threshold values of MS for $C_j$ and $C_2$ are shown with starred and squared dashed lines in Figure 4-3, respectively. As can be seen, MS threshold of $C_2$ is about 0.2 dB less than that of $C_j$. This is justifiable by noticing that MS is the limit of BP with infinite SNR overestimation (see Chapter 3), and that $C_2$ is more robust against SNR mismatch than $C_j$ is. Moreover, Figure 4-3 shows that although MS is robust against mismatch, it experiences more than 1.2 dB and 0.95 dB performance loss compared to BP at offset zero for $C_j$ and $C_2$, respectively.

![Figure 4-3: BP(-) and MS(-->) thresholds for different irregular ensembles.](image-url)
Using the discussions in Chapter 3, if one is interested in designing codes that are robust against mismatch in the unlimited positive range of offsets, one should optimize them for the MS algorithm. The design of optimal irregular ensembles for MS using density evolution is briefly discussed in [36]. The following rate-1/2 ensemble with $D_v=20$ has been optimized for MS in [8][36].

$$\lambda_j(x) = 0.303702x + 0.277538x^2 + 0.028432x^5 + 0.200133x^6 + 0.190195x^{10},$$

$$\rho_j(x) = 0.016x^5 + 0.984x^6.$$

We denote this ensemble by $C_3$. This is the same ensemble as the one referred to as $C'$ in Section 4.1. The BP and MS threshold values of $C_3$ are shown in Figure 4-3 by circled full and dashed lines, respectively. As can be seen, $C_3$ is very robust against SNR overestimation and its threshold value increases only by about 0.2 dB when mismatch increases from zero to infinity. The corresponding increase for $C_1$ is about 1.2 dB. Compared to $C_1$, the maximum threshold of $C_3$ in the offset range $[0, +\infty]$, which is the same as the MS threshold, is decreased by about 0.6 dB. Ensemble $C_3$ also outperforms $C_1$ for any offset value outside the range of $[-2\text{dB}, 2\text{dB}]$. There is however a relatively large penalty of about 0.4 dB at offset zero for the improved sensitivity to mismatch. Compared to the designed ensemble $C_2$, ensemble $C_3$ is inferior throughout the mismatch range of $[-2\text{dB}, 2\text{dB}]$. The performance loss of $C_3$ compared to $C_2$ over this range is between 0.15 dB, at $\eta = -2$ dB, and 0.3 dB, at $\eta = 0$ dB.
Figure 4-4: BER performance of $C_1$ and $C_2$ for SNR values of 1 dB and 1.3 dB at different offset values.

It should be noted that as an efficient replacement for the density evolution approach of [8][36], one can use the EXIT chart approach proposed in this chapter to optimize irregular ensembles for MS. In this case, the EXIT charts are generated for $\eta = +\infty$.

As another example, consider the offset range of interest is [0 dB, 3 dB]. We obtain the following degree distributions for a rate-$1/2$ ensemble with $D_v = 20$ ($C_4$):

$$\lambda_4(x) = 0.258x + 0.270x^2 + 0.011x^5 + 0.189x^6 + 0.272x^{19},$$

$$\rho_4(x) = 0.320x^5 + 0.680x^7.$$

The BP and MS thresholds for this ensemble are shown in Figure 4-3 by crossed solid and dashed lines, respectively. The maximum BP threshold of $C_4$ is about 0.35 dB lower than that of $C_1$ over the offset range of interest. This is at the expense of about 0.2 dB performance loss at zero offset.
Our examples demonstrate a trade-off between the BP performance of an ensemble at offset zero and its robustness against SNR mismatch. The more robust an ensemble is, the larger its BP threshold at offset zero becomes.

For finite-length simulations, we construct two codes of length 50,000 based on the degree distributions of $C_1$ and $C_2$. We refer to these codes by the same name as their corresponding ensembles. The Tanner graphs for both codes are free of cycles of length 4. The BER values of both codes versus SNR offset are reported in Figure 4-4 for SNR values of 1 dB and 1.3 dB. The maximum number of iterations is set to 100, and for each simulation point, at least 100 codeword errors are generated. In the figure, we have also given the asymptotic values, corresponding to the waterfall regions of the curves, obtained from Figure 4-3, by dashed lines. As can be seen, these asymptotic results are consistent with the simulation results. Figure 4-4 shows that $C_2$ is more robust against offset than $C_1$ is. For example, if our aim is to maintain a BER level below $10^{-4}$, for SNR of 1 dB, $C_1$ is only applicable in the offset range of $[-1.5\,\text{dB},\,1.5\,\text{dB}]$ while $C_2$ can tolerate mismatch values in the range of $[-1.7\,\text{dB},\,2.0\,\text{dB}]$. The difference becomes larger if we have a small increase of 0.3 dB in SNR; $C_1$ can then tolerate mismatch values in the range of $[-2.2\,\text{dB},\,2.7\,\text{dB}]$ while this range is $[-2.4\,\text{dB},\,4\,\text{dB}]$ for $C_2$. These advantages are obtained with basically no loss in performance at offset zero. Figure 4-5 shows the BER performance of both codes at offset zero. As can be seen, both codes perform almost identically.

4.5 Some important notes

It is worthwhile to mention that we do not claim that by using the designed codes, there will be no need for a channel estimator. Depending on the channel specifications
and the range and the rate of change for SNR, one may choose to use a certain channel estimation method. Any channel estimator however would introduce some error in estimating the channel (there is no perfect estimation). In fact, that is why there has been research on the effects of SNR mismatch on the performance of BP. Also depending on how fast the SNR of the channel changes, it may take the estimator a while before it can provide a rather precise estimation of the channel. Under these circumstances, one may choose to design codes that are robust against the errors in channel estimation. Note that our codes outperform the conventional codes in the min-max sense as long as there is error in SNR estimation (even though small). Moreover using the robust codes does not add any extra complexity to the system. So the robustness is achieved with no added complexity.

Figure 4-5: BER performance of $C_1$ and $C_2$ at $\eta=0$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.6 Concluding remarks

In this chapter, using EXIT charts, we designed irregular LDPC codes that are robust against channel estimation errors, when decoded by the belief propagation algorithm. In a min-max sense, the designed codes outperform irregular codes, conventionally optimized for zero SNR mismatch, over the SNR offset range of interest. The performance improvement over the conventional irregular codes in many cases is maintained even outside the mismatch range of interest. We also confirm our asymptotic results, obtained by EXIT charts, using finite-length simulations. The EXIT charts generated in this work can also be used for the design of irregular ensembles optimized for the min-sum algorithm. This would be an efficient replacement for the more complex density evolution approach.
Chapter 5 Deterministic Methods for Designing LDPC codes over Binary Erasure Channels

5.1 Introduction

For analysis and design of LDPC codes, among different binary-input symmetric-output channels, the Binary Erasure Channel has been widely considered as the initial step due to the simplicity of the decoder structure. In [20][44][45], a complete mathematical analysis for the performance of LDPC codes over the BEC, both
asymptotically and for finite block lengths, has been developed. For other types of channels such as BSC and the BIAWGN channel, only asymptotic analysis is available [6]. For a variety of channels, the search for the best ensemble can be carried out based on different asymptotic analysis tools such as density evolution [7][29] and Extrinsic Information Transfer (EXIT) charts [8] through an optimization process. In [20], a linear programming approach based on density evolution is used to find good degree distributions for the BEC. For the code design, there are two main categories in general:

1) For a given channel parameter, we look for a code with maximum rate and negligible probability of error or erasure;

2) For a given rate, the code capable of providing a reliable transmission for the worst possible channel parameter is designed.

The second category is of more practical interest, while the first category is usually easier to design. For a given set of constituent variable and check node degrees, and for a given BEC parameter $e$ (a given code rate $R$), the ensemble $(\rho(x), \lambda(x))$ which provides the highest reliable transmission rate (highest erasure protection) is called the optimum ensemble.

Optimization-based design methods are computationally expensive especially when a large number of constituent variable and check node degrees are permitted in the optimization process. In this chapter, our aim is to deterministically design a close-to-optimum ensemble for a given check node degree distribution and a given number $P$ of constituent variable node degrees. The designed ensembles are expected to perform closely to the best ensembles designed by optimization. For both categories of code design, we consider two cases:
A) The case where all the variable node degrees from 2 to a maximum degree $N$ are available ($P = N-1$);

B) the case where not all the degrees from 2 to $N$ are used ($P \neq N-1$)

The ensembles designed in the two scenarios are referred to as Type-A and Type-B ensembles\(^{10}\), respectively. In practice, the choice of $P$ may be affected by implementation considerations, where smaller values would be preferred. Although in this chapter we focus on the design of ensembles for a given check node degree distribution, the designed ensembles can also be used to optimize both the variable node and the check node degree distributions iteratively in an optimization loop. In each iteration, $\rho(x)$ and subsequently $\lambda(x)$ (obtained by the method proposed in this chapter), is modified to optimize the cost function (rate or threshold).

In [21][22][46][47], the authors introduce sequences of degree distributions that asymptotically achieve the capacity of a BEC for large values of maximum variable and check node degrees. For finite values of maximum variable and check node degrees, those sequences can also be used to deterministically design LDPC codes over a BEC. In fact, the constructions of [21][22][46][47] are a subset of constructions discussed in this chapter (Type-A in Category 2 of code design). Here, we show that more favorable solutions for finite values of $P$ do exist in our extended family of designs, i.e., for a given rate, a given check node degree distribution and a given $\epsilon > 0$, the designed ensemble can have a threshold in $\epsilon$-neighborhood of the capacity upper bound with a smaller value of $P$ and a smaller maximum variable node degree, compared to the ensembles of [21][22][46][47]. It should be noted that although the sequences of [21][22][46][47] are

\(^{10}\) We note that Type-A and Type B ensembles for the first and second category of code design correspond to GC and GR ensembles of Chapter 6 respectively.
special cases of the designs proposed in this chapter, the approach taken here to derive them is different and much simpler than that of [21][22][46][47]. In addition, it has been proved in Chapter 6 that for large values of maximum variable and check node degrees, all the designed ensembles (in both categories and for both types) are capable of achieving the capacity of the BEC.

The chapter is organized as follows. In Section 5.2, we discuss the first category of code design and prove a few lemmas that are used in the design process. Section 5.3 generalizes the results of Section 5.2 to the second category of code design. In Section 5.4, we provide some design examples. Section 5.5 concludes the chapter. The proofs of all the lemmas and theorems of this chapter are given in Appendix A.

5.2 Code design for the highest rate

In this section, we consider the case where we are given a check node degree distribution $\rho(x)$ and a certain channel erasure probability $e$. Our goal is to find the variable node degree distribution $\lambda(x)$ of a convergent ensemble with the largest rate. If $N$ ($\geq 3$) denotes the maximum variable node degree, it is apparent from (2-11) that we need to minimize the average variable node degree or maximize its inverse:

$$\overline{d}_v^{-1} = \sum_{i=2}^{N} \lambda_i / i.$$  \hspace{1cm} (5-1)

In fact, the optimization of the rate is equivalent to maximizing $\overline{d}_v^{-1}$, subject to two constraints, equation (2-8) and inequality (2-21) which guaranties the code's convergence. From (5-1), it can be seen that in order to maximize $\overline{d}_v^{-1}$, higher percentages
have to be assigned to lower degree variable nodes. The following lemma is a formulation of this idea:

**Lemma 5-1**: Consider a given check node degree distribution, a given channel parameter and a given set of constituent variable node degrees. Let $C$ be a convergent code ensemble of rate $R$ with variable node degree distribution $\lambda(x) = \sum_{i=2}^{N} \lambda_i x^{i-1}$. For given integer numbers $a$ and $b$ in the interval $[2, N]$, $a \neq b$, we form a new ensemble $C'$ with rate $R'$ such that $\lambda'_a = \lambda_a - k$, $\lambda'_b = \lambda_b + k$ and $\lambda'_i = \lambda_i$, for $i \not\in \{a, b\}$ (k is chosen such that $\lambda'_a > 0$ and $\lambda'_b \leq 1$). We then have:

1) If $a > b$, then $R' > R$.

2) If $a < b$, then $C'$ is convergent.

The first part of this lemma proposes a general approach to increase the rate but does not guarantee the convergence of the resulting ensemble. In fact, in conventional code optimization methods, the convergence of any newly constructed ensemble has to be verified by testing (2-21). In what follows, we derive upper bounds on $\lambda_i$ values based on the convergence condition (2-21) in the vicinity of $x = 0$. Then using these bounds, we can construct close-to-optimum ensembles whose convergence is ensured and need not to be checked by (2-21). To get such upper bounds, we consider the Taylor expansion of $\rho^{-1}(1-x)$. It can be shown that if $\rho(x)$ is a degree distribution, the Taylor series of $\rho^{-1}(1-x)$ around $x = 0$ is convergent [21]. Let

$$\rho^{-1}(1-x) = 1 - \sum_{i=2}^{\infty} T_i x^{i-1}, T_i > 0.$$  \hspace{1cm} (5-2)

By replacing (5-2) in (2-21), we obtain
\[(\varepsilon \lambda_2 - T_2)x + (\varepsilon \lambda_3 - T_3)x^2 + \ldots + (\varepsilon \lambda_N - T_N)x^{N-1} - \sum_{i=N+1}^{\infty} T_i x^{i-1} < 0, \quad 0 < x \leq 1.\] (5-3)

If \(x\) tends to zero, all the terms with powers greater than one can be neglected compared to the first term on the left hand side of (5-3). Therefore, as \(x\) tends to zero, we must have:

\[\frac{(\varepsilon \lambda_2 - T_2)x \leq 0}{\Rightarrow} \lambda_2 \leq \frac{T_2}{\varepsilon}, \quad (5-4)\]

This is the upper bound on \(\lambda_2\). One can verify that \(T_2 = 1/\rho'(1)\) and thus (5-4) is the well-known stability condition \(\varepsilon \lambda_2 \rho'(1) \leq 1\) [7]. Now suppose that we set \(\lambda_2\) equal to the upper bound of (5-4). Then, the first term on the left hand side of (5-3) becomes zero. In this case, as \(x\) tends to zero, the term with \(x^2\) becomes dominant and the necessary condition for convergence is

\[\frac{(\varepsilon \lambda_3 - T_3)x^2 \leq 0}{\Rightarrow} \lambda_3 \leq \frac{T_3}{\varepsilon}.\]

We can continue in a similar fashion and obtain an upper bound on \(\lambda_i\), i.e., \(\lambda_i \leq T_i / \varepsilon\), for \(3 \leq i \leq N - 1\), assuming that all \(\lambda_j\) values for \(j = 2, \ldots, i-1\), have their maximum values.\(^{11}\)

5.2.1 Type-A ensembles

Now assume that all variable node degrees from 2 to \(N\) are available. The above inequalities suggest that for a given \(\varepsilon\) and a given \(\rho(x)\), the following ensemble, which is designed deterministically, could be a close-to-optimum candidate if it is convergent:

\[\lambda_i = \frac{T_i}{\varepsilon}, \quad 2 \leq i \leq N - 1; \quad \lambda_N = 1 - \sum_{i=2}^{N-1} \lambda_i, \quad (5-5)\]

\(^{11}\) It should be noted that this result coincides with the flatness condition proposed in [46] for capacity achieving sequences. The sequences of [46] however belong to the second category of code design.
We show that for a given $\rho(x)$ and a given $\varepsilon$, there exists a lower bound on $N$ that will ensure the convergence and an upper bound which guarantees $\lambda_N$ to be positive. The following lemma indicates that a unique $N$ satisfies both conditions. We call the corresponding degree distributions Type-A.

**Theorem 5-1**: Consider a given check node degree distribution $\rho(x)$, and denote the $i$th term of the Taylor expansion of $\rho^{-1}(1-x)$ at $x = 0$ by $T_i$, as in (5-2). For a given channel parameter $\varepsilon \geq T_2$ and a set of constituent variable node degrees from 2 to $N$ ($N > 2$), there exists a unique $N$ that satisfies the following bounds:

$$\sum_{i=2}^{N} T_i > \varepsilon, \tag{5-6}$$

$$\varepsilon \geq \sum_{i=2}^{N-1} T_i. \tag{5-7}$$

For such $N$, the convergence of the Type-A ensemble is ensured and $\lambda_N \geq 0$.

Note that if we would like to design a code for a channel parameter $\varepsilon$ which is less than $T_2$, we have to decrease $T_2$ by increasing the average check node degree through the modification of $\rho(x)$.

**Theorem 5-2**: Consider the Type-A ensemble $C$ designed based on (5-5) for a given channel parameter $\varepsilon$. The channel parameter $\varepsilon$ is then the threshold of $C$.

**Example 5-1**: For $\varepsilon = 0.48$ and $\rho(x) = x^5$, it can be seen that the value of $N$ which satisfies (5-6) and (5-7) is $N = 13$. The variable node degree distribution for the Type-A ensemble is:

Note that based on (5-1) and (5-5), the condition $\varepsilon \geq T_2$ is satisfied for LDPC ensembles with variable node degrees equal to or larger than two. Almost all practical ensembles have this property.
\[
\lambda(x) = 0.4167x + 0.1667x^2 + 0.1000x^3 + 0.0700x^4 + 0.0532x^5 + 0.0426x^6
+ 0.0353x^7 + 0.0300x^8 + 0.0260x^9 + 0.0229x^{10} + 0.0204x^{11} + 0.0165x^{12}
\]

This ensemble has rate \( R = 0.4998 \) and its threshold is 0.48.

Note that Type-A ensembles are optimal in a greedy sense, in that, starting from degree-2 variable nodes, we maximize the percentage of edges connected to lower degree variable nodes and thus aim for maximizing the rate of the ensemble. For a fixed check degree distribution and a given channel parameter, however, the value of \( N \) and thus the number of constituent variable node degrees are both fixed and dictated by Theorem 5-1. In the following, we introduce new ensembles, where we have the flexibility to determine the number of constituent variable node degrees \( P \) and design ensembles with \( P < N-1 \). The cost associated with reducing \( P \) is a reduction in rate.

5.2.2 Type-B ensembles

Given a check node degree distribution and a channel parameter, Theorem 5-1 indicates that there exists a unique \( N \) that satisfies (5-6) and (5-7). For such \( N \), consider a variable node degree distribution which includes a few consecutive degrees starting from 2 and ending at \( P < N \) and the maximum variable node degree \( N \).

Example 5-2: Consider the Type-A ensemble of Example 5-1, in which \( D_c = 6 \), and the maximum variable node degree \( N \) is 13. For \( P = 4 \), the new ensemble has variable node degrees 2, 3, 4 and 13.

For such constituent variable node degrees, a Type-B ensemble is constructed based on:

\[
\lambda_i = \frac{T_i}{\epsilon}, \quad 2 \leq i \leq P; \quad \lambda_N = 1 - \sum_{i=2}^{P} \lambda_i.
\]
where $T_i$ is defined in Theorem 5-1.

Since $N$ satisfies the conditions of Theorem 5-1, part 2 of Lemma 5-1 will ensure the convergence of the new ensemble.

Consider now a Type-B ensemble with variable node degrees 2, 3, ..., $P$, and a maximum variable node degree $N$. Based on part 1 of Lemma 5-1, if instead of $N$ we choose a smaller maximum variable node degree $D_v$ with the same percentage of adjacent edges, the newly constructed ensemble has a higher rate but can be non-convergent. By choosing the smallest $D_v$ which results in a convergent ensemble, we can create a new ensemble, referred to as Modified Type-B or Type-MB. Note that the variable node degree distribution for this ensemble is the same as that of Type-B ensemble with $\lambda_o$ replacing $\lambda_N$. Also note that with an argument similar to that of Theorem 5-2, we can show that the thresholds of both Type-B and Type-MB ensembles are equal to the channel parameter.

Example 5-3: Consider the Type-B ensemble of Example 5-2 with $P = 4$. This ensemble has variable node degrees 2, 3, 4 and 13, with coefficients $\lambda_2 = 0.4167, \lambda_3 = 0.1667, \lambda_4 = 0.1000$ and $\lambda_{13} = 0.3176$, respectively. The rate of this ensemble is $R = 0.4679$, which is less than the rate of the Type-A ensemble of Example 5-1, as expected. If we keep decreasing the maximum variable node degree, we see that a degree distribution with degrees 2, 3, 4 and 8 is convergent while one with 2, 3, 4 and 7 is not. Therefore, for $\varepsilon = 0.48$ and $\rho(x) = x^3$, the Type-MB ensemble has variable node degrees 2, 3, 4 and 8 with coefficients $\lambda_2 = 0.4167, \lambda_3 = 0.1667, \lambda_4 = 0.1000$ and $\lambda_8 = 0.3176$. This ensemble has a rate $R = 0.4926$ which is in between the rates of Type-A and Type-B ensembles, and in fact very close to the rate of the Type-A ensemble. It is however important to note that
compared to the Type-A ensemble, which has 12 different variable node degrees with a maximum degree of 13, this ensemble has only 4 different variable node degrees and the maximum degree is only 8.

In the following proposition, we derive a lower bound on $D_v$, which is a sufficient condition for convergence.

**Proposition 5-1:** Consider an ensemble $C$ with a given check node degree distribution and a set of consecutive constituent variable node degrees from 2 to $P$ and a maximum variable node degree $D_v$ ($D_v \geq P+1$). Suppose that the channel parameter $\epsilon$ is given and that $N$ is computed based on Theorem 5-1. Let $T_i$ be the $i$th term of the Taylor expansion of $\rho^{-1}(1 - x)$. For ensemble $C$, also let $\lambda_i = T_i / \epsilon$, $2 \leq i \leq P$; and $\lambda_{P+1} = 1 - \sum_{i=2}^{P} \lambda_i$. Then the following lower bound on $D_v$ is a sufficient condition for the convergence of $C$:

$$D_v \geq N - \left( \sum_{i=P+1}^{N} (N-i)T_i \right) / (\epsilon - \sum_{i=2}^{P} T_i).$$

(5-8)

For the ensemble of Example 5-3, the lower bound of (5-8) is equal to 7.8590, which suggests choosing $D_v = 8$. In this case, 8 is in fact the smallest possible value for $D_v$. In general however the lower bound of (5-8) may not result in the smallest possible answer for $D_v$. Nevertheless, one can use this lower bound as a starting point to conduct a quick search for the smallest $D_v$, which results in a convergent ensemble.

### 5.3 Code design for the highest threshold

In this section, we are interested in designing $\lambda(x)$ for an ensemble $C$ with a given check node degree distribution $\rho(x)$ and a certain rate $R$ that has the largest possible
threshold. Suppose that the largest threshold is equal to $\varepsilon$, and is achieved by ensemble $C$. This implies that for the channel parameter $\varepsilon$, ensemble $C$ has the highest rate which is equal to $R$. This in turn suggests that a similar approach as the one described in Section 5.3 can also be applied to designing close-to-optimum ensembles for a given rate.

### 5.3.1 Type-A ensembles

For a given $\rho(x)$ and a given rate $R$, for Type-A ensembles, we consider the case where all variable node degrees from 2 to a maximum degree $N$ are available ($N$ will be determined later). Suppose that the threshold is equal to $\varepsilon$. We can then use (5-5) to compute the $\lambda_i$ values based on $\varepsilon$. We then use (2-11) to obtain the following equation:

$$R = 1 - \frac{\bar{d}_e^{-1}}{1/N + \frac{1}{\varepsilon} \sum_{i=2}^{N-1} T_i(1/i - 1/N)}.$$  \hfill (5-9)

Solving this equation for $\varepsilon$ results in

$$\varepsilon = \frac{\sum_{i=2}^{N-1} T_i(1/i - 1/N)}{\sum_{i=2}^{N-1} T_i(1/i - 1/N)} = \frac{\sum_{i=2}^{N-1} T_i(1/i - 1/N)}{(1-R)^{-1} - 1/N}.$$  \hfill (5-10)

The variable node degree distribution can then be computed as:

$$\lambda_i = T_i \frac{(1-R)^{-1} - 1/N}{\sum_{i=2}^{N-1} T_i(1/i - 1/N)} \quad 2 < i \leq N - 1; \lambda_N = 1 - \sum_{i=2}^{N-1} \lambda_i.$$  \hfill (5-11)

Now, for the ensemble to converge, it has to satisfy (5-6):

$$\sum_{i=2}^{N} T_i > \frac{\sum_{i=2}^{N-1} T_i(1/i - 1/N)}{d_v^{-1} - 1/N}$$

and thus:
Also, (5-7) should hold for the $N^{th}$ coefficient to be non-negative:

$$\overline{d}_{v}^{-1} \sum_{i=2}^{N} T_i \leq \sum_{i=2}^{N} T_i / i.$$  \hspace{1cm} (5-13)

**Theorem 5-3:** For a given code rate $R$ and a given check node degree distribution (and thus a given $\overline{d}_{v}^{-1}$), if $R < 1 - 2 / \overline{d}_{c}$,\(^{13}\) then there exists a unique value of $N$ that satisfies (5-12) and (5-13).

Note that if the code rate $R$ does not satisfy the inequality of Theorem 5-3, we would have to increase $\overline{d}_{c}$ through modifying $\rho(x)$.

To summarize the design: For a given rate and a given check node degree distribution, we first find $T_i$ values, then compute $N$ from (5-12) and (5-13). Coefficients $\lambda_i$ are finally obtained based on (5-11). Note that with an argument similar to that of Theorem 5-2, one can show that the channel parameter obtained by (5-10) is in fact the true threshold of the Type-A ensemble.

**Example 5-4:** For $R = 0.5$ and $\rho(x) = x^5$, it can be seen that the value of $N$ which satisfies (5-12) and (5-13) is 13. The variable node degree distribution for the Type-A ensemble is:

\(^{13}\) Note that this inequality is satisfied for any ensemble whose variable node degrees are at least two.
\[
\lambda(x) = 0.4169x + 0.1667x^2 + 0.1000x^3 + 0.0700x^4 + 0.0532x^5 + 0.0426x^6 \\
+ 0.0353x^7 + 0.0300x^8 + 0.0260x^9 + 0.0229x^{10} + 0.0204x^{11} + 0.0160x^{12}.
\]

This code has a threshold \( \varepsilon = 0.4798 \).

### 5.3.2 Type-B ensembles

Similar to Type-B ensembles of Section 5.3, in this section, we consider ensembles with a few consecutive variable node degrees from 2 to \( P \) and a maximum degree \( N \). We initiate the design by computing \( N \) from (5-12) and (5-13) and then computing \( \varepsilon \) from the following equation:

\[
\varepsilon = \frac{\sum_{i=2}^{P} T_i (1/i - 1/N)}{(1 - R)^{-1} d_e^{-1} - 1/N}.
\]  

(5-14)

We then set

\[
\lambda_i = T_i / \varepsilon, 2 \leq i \leq P, \lambda_N = 1 - \sum_{i=2}^{P} \lambda_i.
\]

(5-15)

**Theorem 5-4**: Coefficient \( \lambda_N \) in (5-15) is positive.

In the following theorem, we also show that the new sequence is convergent.

**Theorem 5-5**: A Type-B ensemble of rate \( R \) is convergent over a channel with parameter equal to the value given in (5-14). Moreover, this value is the threshold of the Type-B ensemble.

Similar to the case in Section 5.3, to obtain a better threshold for a given \( P \), we can design a Type-MB ensemble by decreasing the maximum variable node degree from \( N \) to a smaller value \( D_v \). To design this ensemble, we use (5-14) and (5-15) with \( N \) replaced by \( D_v \), and find the smallest \( D_v \) for which the ensemble is convergent for the
channel parameter given by (5-15). With a similar argument used for Type-A and B ensembles, the value of (5-15) is the threshold of the Type-MB ensemble.

Unfortunately, for this category of code design, we have not been able to obtain a lower bound for $D_v$ similar to that of Proposition 5-1. One however can perform a maximum of $N-P$ trials to find the smallest $D_v$. Each trial consists of computing $\varepsilon$ from (5-14) and $\lambda(x)$ from (5-15), where in both equations $N$ is replaced by $D_v$. We then need to check whether inequality (2-21) holds for the tested $D_v$.

**Example 5-5:** Consider the Type-B ensemble with $P = 4$ and $D_c = 6$ corresponding to the Type-A ensemble of Example 5-4. This ensemble has variable node degrees 2, 3, 4 and 13, with coefficients $\lambda_2 = 0.4521$, $\lambda_3 = 0.0808$, $\lambda_4 = 0.1085$ and $\lambda_{13} = 0.258$, respectively. The threshold of this ensemble computed by (5-14) is $\varepsilon = 0.4424$, which is less than that of the Type-A ensemble of Example 5-4, as expected. If we keep decreasing the maximum variable node degree, we see that a degree distribution with degrees 2, 3, 4 and 8 is convergent while one with 2, 3, 4 and 7 is not. Therefore, for $R = 0.5$ and $\rho(x) = x^5$, the Type-MB ensemble has variable node degrees 2, 3, 4 and 8 with coefficients $\lambda_2 = 0.4266$, $\lambda_3 = 0.1706$, $\lambda_4 = 0.1024$ and $\lambda_8 = 0.3003$. This ensemble has a threshold $\varepsilon = 0.4688$, obtained from (5-14), which is in between the thresholds of Type-A and Type-B ensembles. It is however important to note that compared to the Type-A ensemble, which has 12 different variable node degrees with a maximum degree of 13, this ensemble has only 4 different variable node degrees and the maximum degree is only 8.
5.4 Simulation results

For simulation results, we consider both categories of code design for the highly popular check-regular ensembles\textsuperscript{14}. We first consider an upper bound from [21] which is useful to measure the performance of our designed ensembles: For a given rate $R$ and a given average check node degree $d_c$, the best achievable threshold is upper bounded by

$$
3 = (1-R)(1-R^{d_c}).
$$

By modifying the upper bound of [21], for a given channel parameter $\varepsilon$ and a given $d_c$, we obtain the following upper bound on the best achievable rate:

$$
\mathcal{R} = 1 - \frac{\varepsilon}{(1-(1-\varepsilon)^{d_c})}.
$$

**Example 5-6:** Consider the second category of code design for rate one half. Suppose that there is a constraint of $P = 4$ on the number of different variable node degrees. We consider check-regular ensembles with $D_c = 5, 6,$ and $7$. Table 5-1 shows the designed Type-MB ensembles for each check node degree. Note that for $R = 0.5$, the capacity upper bound implies $\varepsilon < 1 - R = 0.5$. For each ensemble, we have shown the ratio of the threshold to $1-R = 0.5$ as well as to $3$.

As can be seen, the threshold improves by increasing $D_c$ and the ensemble with check node degree $7$ achieves close to $97\%$ of the upper bound $3$. In fact for $P = 4$, this is the best threshold than can be obtained by Type-MB ensembles. By increasing $D_c$ further, the threshold decreases unless we allow $P$ to also increase.

**Example 5-7:** In this example, by still focusing on check-regular ensembles, we allow $P$ to take values between 5 and 10 and for each value of $P$, we find the value of $D_c$ which

\textsuperscript{14} The popularity of check-regular ensembles is due to their better erasure correcting capabilities and simpler implementation in hardware. For a reference on the former, see [22].
results in the best threshold for Type-MB ensembles. These results are reported in Table 5-2.

Table 5-1: Type-MB ensembles for $R = 0.5$ and $P = 4$.

<table>
<thead>
<tr>
<th>$D_c$</th>
<th>$D_v$</th>
<th>$\varepsilon_{MB}/0.5$</th>
<th>$\varepsilon_{MB}/3$</th>
<th>$\lambda(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>0.8873</td>
<td>0.9159</td>
<td>$0.5635x + 0.2113x^2 + 0.1233x^3 + 0.1019x^4$</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.9376</td>
<td>0.9525</td>
<td>$0.4266x + 0.1706x^2 + 0.1024x^3 + 0.3003x^4$</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>0.9610</td>
<td>0.9686</td>
<td>$0.3459x + 0.1445x^2 + 0.883x^3 + 0.4203x^4$</td>
</tr>
</tbody>
</table>

Table 5-2: Best Type-MB check-regular ensembles for $R = 0.5$ and different values of $P$.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$D_c$</th>
<th>$D_v$</th>
<th>$\varepsilon_{MB}/0.5$</th>
<th>$\varepsilon_{MB}/3$</th>
<th>$\lambda(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>12</td>
<td>0.9624</td>
<td>0.9700</td>
<td>$0.3464x + 0.1443x^2 + 0.0882x^3 + 0.0625x^4 + 0.3587x^5$</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>13</td>
<td>0.9716</td>
<td>0.9793</td>
<td>$0.3431x + 0.1429x^2 + 0.0874x^3 + 0.0619x^4 + 0.0474x^5 + 0.3173x^6$</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>14</td>
<td>0.9761</td>
<td>0.9838</td>
<td>$0.3415x + 0.1423x^2 + 0.0870x^3 + 0.0616x^4 + 0.0472x^5 + 0.0380x^6 + 0.2824x^7$</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>15</td>
<td>0.9783</td>
<td>0.9860</td>
<td>$0.3407x + 0.1420x^2 + 0.0868x^3 + 0.0615x^4 + 0.0471x^5 + 0.0380x^6 + 0.0316x^7 + 0.2524x^8$</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>22</td>
<td>0.9836</td>
<td>0.9875</td>
<td>$0.2905x + 0.1245x^2 + 0.0771x^3 + 0.0550x^4 + 0.0425x^5 + 0.0344x^6 + 0.0288x^7 + 0.0247x^8 + 0.0327x^9$</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>23</td>
<td>0.9864</td>
<td>0.9902</td>
<td>$0.2897x + 0.1241x^2 + 0.0768x^3 + 0.0549x^4 + 0.0423x^5 + 0.0343x^6 + 0.0287x^7 + 0.0246x^8 + 0.0215x^9 + 0.0303x^10$</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Example 5-8: In this example, we compare the performance of the Type-MB ensemble with $D_c = 7$ and $P = 8$ designed in Example 5-7 with its corresponding Type-A ensemble with the same check node degree distribution. The Type-A ensemble has the following variable node degree distribution:

$$\lambda(x) = 0.3394x + 0.1414x^2 + 0.0864x^3 + 0.0612x^4 + 0.0469x^5 + 0.0378x^6 + 0.0315x^7 + 0.0269x^8 + 0.0234x^9 + 0.0207x^{10} + 0.0185x^{11} + 0.0167x^{12} + 0.0152x^{13} + 0.0139x^{14} + 0.0128x^{15} + 0.0119x^{16} + 0.0111x^{17} + 0.0104x^{18} + 0.0097x^{19} + 0.0092x^{20} + 0.0087x^{21} + 0.0082x^{22} + 0.0078x^{23} + 0.0074x^{24} + 0.0071x^{25} + 0.0067x^{26} + 0.0065x^{27} + 0.0025x^{28}$$

The threshold of this ensemble is equal to 0.4910 which is slightly better than that of Type-MB ensemble (0.4891). This is at the expense of 28 different variable node degrees (instead of 8 for Type-MB) and the maximum variable node degree of 29 (instead of 15 for Type-MB). In Figure 5-1, we have shown the finite block length simulation results for the two codes. The block length is selected to be 5000 and the maximum number of iterations is limited to 200. For each simulation point, one hundred codeword erasures are generated. As can be seen, the two codes perform closely in finite block length too.

In the next example, we demonstrate that for large enough maximum variable and check node degrees and $P$, Type-MB ensembles can practically achieve the capacity similar to the sequences of [21][22][46][47]. In fact, we have been able to rigorously prove this result in Chapter 6.
Example 5-9: We consider again rate one-half check-regular ensembles. With $D_c = 11$ and $P = 90$, we obtain $D_v = 203$ for a Type-MB ensemble. The threshold for this ensemble is equal to 0.4993 which is 99.9% of the capacity upper bound. If we use the check regular sequences of [22] with $D_c = 11$, to achieve the same percentage of the capacity bound, the designed code has to have 522 different constituent variable node degrees. Also, the maximum variable node degree increases from 203 to 523.

In the following, we compare our results with those obtained by optimization.

Example 5-10: From the database [48] of optimized LDPC codes, we consider the following check-regular rate one-half ($R = 0.5$) ensemble $C_J$ with $D_c = 7$:

$$
\lambda_{C_j}(x) = 0.3354x + 0.11716x^2 + 0.0095x^3 + 0.0783x^4 \\
+ 0.1620x^5 + 0.1305x^6 + 0.1126x^7
$$
For this ensemble, $P = 7$ and the threshold $\varepsilon_{C_1}$ is equal to 0.4917. For $P = 7$ and $D_v = 7$, the Type-MB ensemble $C_2$ has $D_v = 14$ with the following variable node degree distribution:

$$\lambda_{C_2}(x) = 0.3415x + 0.1423x^2 + 0.0870x^3 + 0.0616x^4$$
$$+ 0.0472x^5 + 0.0380x^6 + 0.2824x^{13}$$

This ensemble has a threshold $\varepsilon_{C_2} = 0.4880$ which is only slightly less than that of $C_1$. It should however be noted that the maximum variable node degree for $C_2$ is smaller than that of $C_1$. If we restrict the maximum variable node degree to 14 or smaller, we find another optimized ensemble $C_3$ in [48] with the following variable node degree distribution:

$$\lambda_{C_3}(x) = 0.2853x + 0.3135x^2 + 0.1162x^3 + 0.2848x^{12}.$$ Incidentally, this ensemble has the same threshold $\varepsilon_{C_2} = 0.4880$ as $C_2$. The advantage of this ensemble over $C_2$ is however the fewer number of constituent variable node degrees and the maximum variable node degree of 13 instead of 14.

**Example 5-11:** In this example, we consider the first category of code design. Let $D_v = 5$, $\varepsilon = 0.48$, and consider a constraint of $P = 4$ on the number of different variable node degrees. Using (5-6) and (5-7), we obtain $N = 7$. If we limit our search for an optimal ensemble to the maximum variable node degree of 7, there are a total of \( \binom{6}{4} = 15 \) different sets of constituent variable node degrees to be tested. We have performed optimization using exhaustive search for each set of constituent variable node degrees. Some of the best results are given in Table 5-3. Note that for each row of Table 5-3, the best $\lambda(x)$ which maximizes the rate for a convergent ensemble has been obtained. The
highest ratio $R / \mathcal{R}$ in the table is 0.9624, which corresponds to achieving about 96% of the rate upper bound.

Using our design method, we obtain a Type-MB ensemble with the following variable node degree distribution:

$$\lambda(x) = 0.5208x + 0.1953x^2 + 0.1139x^3 + 0.1699x^5.$$  

The rate of this ensemble is 0.4769 which is more than 95% of the upper bound $\mathcal{R}$. This is only slightly worse than the best result from the exhaustive search. The complexity of our design however is substantially lower than that of the exhaustive search.

Table 5-3: Results for rate optimization of LDPC code ensembles with $D_c = 5$ and 4 different constituent variable node degrees at channel erasure probability 0.48.

<table>
<thead>
<tr>
<th>Variable node degrees</th>
<th>$R$</th>
<th>$R/\mathcal{R}$</th>
<th>$\lambda(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2,3,4,6]</td>
<td>0.4821</td>
<td>0.9622</td>
<td>0.5206$x + 0.0730x^2 + 0.4059x^3 + 0.0005x^5$</td>
</tr>
<tr>
<td>[2,3,4,5]</td>
<td>0.4822</td>
<td>0.9624</td>
<td>0.5196$x + 0.1172x^2 + 0.2950x^3 + 0.0682x^4$</td>
</tr>
<tr>
<td>[2,4,5,6]</td>
<td>0.4739</td>
<td>0.9455</td>
<td>0.5208$x + 0.4792x^3$</td>
</tr>
</tbody>
</table>

5.5 Concluding remarks

In this chapter, we propose methods to deterministically design ensembles of irregular LDPC codes for binary erasure channels. The main idea is to maximize the percentage of edges connected to lower degree nodes in a greedy fashion. At finite maximum variable and check node degrees, the designed ensembles perform close to optimal, only slightly inferior to the ensembles designed by exhaustive search (or optimization algorithms with asymptotic analysis tools). This is while the design
complexity of the proposed ensembles is substantially lower than those obtained by exhaustive search.

An important feature of the proposed design is the flexibility to choose the number of constituent variable node degrees $P$, and to deterministically design a close to optimal ensemble under this constraint. Check-regular ensembles designed by this method would provide attractive solutions for implementation.

The proposed ensembles are also asymptotically optimal in that they achieve the capacity as the maximum variable and check node degrees increase. While capacity-achieving ensembles of [22] are special cases of the proposed designs, our derivation approach is different and much simpler. Moreover, our extended family of designs include more attractive solutions for finite values of $P$, i.e., for a given rate, a given check node degree distribution and a given $\varepsilon > 0$, the designed ensembles can have a threshold in $\varepsilon$-neighborhood of the capacity upper bound with a smaller value of $P$ and a smaller maximum variable node degree, compared to the ensembles of [22].

In Chapter 7 we obtain similar results on deterministic designs of irregular LDPC code ensembles for the BIAWGN channel.
Chapter 6 New Sequences of Capacity

Achieving LDPC Code Ensembles over the Binary Erasure Channel

6.1 Introduction

As mentioned in Chapter 5, the ensembles proposed in that chapter can achieve the capacity of BEC for sufficiently large maximum variable node and check node degrees. A similar case has been considered in [21][22][47]. In [21][22][47] for a BEC, a sequence of degree distributions is called capacity achieving of rate $R$ if the largest
possible value of $\epsilon$ that makes the reliable transmission possible (threshold) can be made arbitrarily close to $1-R$ for sufficiently large maximum variable node and check node degrees. Two such sequences, Tornado and right-regular sequences have been introduced in [46] and [21], respectively. In [22], Oswald and Shokrollahi have presented a more general class of capacity achieving sequences that includes Tornado and right-regular sequences. Specifically, they show that right-regular sequences approach the capacity faster than other sequences in the class as the maximum variable node and check node degrees increase. This in turn results in a smaller decoding complexity per iteration to achieve a given performance [22].

The definition for capacity achieving sequences can be extended to the case of fixed channel parameter. More specifically, a sequence of degree distributions is called capacity achieving for the channel parameter $\epsilon$ if the ensemble sequence can provide reliable transmission over a BEC with parameter $\epsilon$ and if the corresponding code rate $R$ can be made arbitrarily close to $1-\epsilon$ for sufficiently large maximum variable node and check node degrees.

In [21][22][47], the constructed ensembles are required to have variable node degree distributions for which all variable node degrees from 2 to a maximum degree $N$ are present. In designing code ensembles to achieve a certain performance, and usually driven by implementation considerations, one may be interested in having smaller maximum variable node and check node degrees as well as smaller variety of constituent variable node and check node degrees. In this chapter based on the ensembles proposed in Chapter 5, we propose a class of sequences of capacity achieving ensembles for which the number of constituent variable node degrees can be reduced to a value $P = [p(N)] < N$,
where \([\cdot]\) denotes the closest integer value. There is a great deal of freedom in choosing \(P\) as the only condition on \(f(N)\) is to be a strictly increasing function of \(N\). The special case where \(P = f(N) = N - 1\) corresponds to the sequences of [21][22][47]. We show that, similar to the sequences of [21][22][47], all the proposed sequences are also capacity achieving. Moreover we demonstrate that for \(P < N - 1\), the proposed sequences are superior to those of [21][22][47] in the sense that for sufficiently large maximum check node and variable node degrees, they can achieve a target performance within a given neighborhood of the capacity upper bound with a smaller maximum variable node degree and smaller number of constituent variable node degrees.

As an example, consider rate one half ensembles with check node degree of 12. Note that the best achievable threshold is upper bounded by \(1 - R = 1/2\). Using the ensemble construction of [21] with the maximum variable node degree 1059 and the variable node degree distribution including all the degrees from 2 to 1059 (1058 different degrees), we can achieve a threshold which is 99.95% of \(1 - R\). The same threshold can be achieved based on our proposed sequences with \(f(N) = \lceil N/4 \rceil + 2\). In this case, the maximum variable node degree is reduced from 1059 to 493 and we only require 267 different variable node degrees instead of 1058 needed for the construction of [21].

To quantify the convergence speed of the proposed sequences as the maximum variable node and check node degrees increase, we prove that with a proper choice of \(f(N)\), our proposed sequences can be asymptotically quasi-optimal [22], i.e., their decoding complexity per iteration increases only logarithmically with the relative increase of the performance with respect to the capacity upper bound. A stronger result for check-regular sequences with \(P = N - 1\) has been established in [22]. In particular, it is
proved in [22] that these sequences are \textit{asymptotically optimal}, i.e., their decoding complexity per iteration increases not only logarithmically with the relative increase of the performance but also the coefficient of this increase, \( \mu \), is equal to one. For the other capacity achieving sequences discussed in [22], including Tornado sequences, we have \( \mu > 1 \). We show that if \( f(N) \) is a linear function of \( N \), our proposed check-regular sequences can be made \textit{asymptotically optimal}.

Due to the fast convergence speed of check-regular sequences and their simpler structure of having only one constituent check node degree, our focus in this chapter will be mainly on check-regular ensembles. The same design principles however can be applied to other capacity achieving sequences including those in [22].

The chapter is organized as follows. In Section 6.2, we introduce our proposed ensembles. In Section 6.3, we prove that they are capacity achieving. We then prove that the proposed sequences can be asymptotically quasi-optimal or asymptotically optimal depending on the choice of \( f(N) \). In Section 6.4, we provide some examples of our sequences and compare their performance with those of [21][22]. Section 6.5 concludes the chapter. The proofs of all lemmas and theorems are given in the Appendix C.

### 6.2 Proposed ensemble sequences

Consider a check node degree distribution \( \rho(x) \). It can be shown that the Taylor series of \( \rho^{-1}(1 - x) \) around \( x = 0 \) is convergent [21]. We thus have

\[
\rho^{-1}(1 - x) = 1 - \sum_{i=2}^{\infty} T_i x^{i-1}, \; T_i > 0.
\]  

(6-1)
We note that for check-regular codes with the check degree $D_c$, there is a closed form expression for $T_i$:

$$T_i = \binom{\alpha}{i-1} (-1)^i, \quad \alpha = 1/(D_c - 1),$$  \hspace{1cm} (6-2)

where the fractional binomial expansion $\binom{\alpha}{i}$ is defined as [21]:

$$\binom{\alpha}{i} = \frac{\alpha(\alpha - 1) \cdots (\alpha - i + 1)}{i!} = \frac{\alpha}{i} \frac{1 - \frac{\alpha}{i - 1}}{1 - \frac{\alpha}{i - 2}} \cdots \frac{1 - \frac{\alpha}{2}}{1 - \alpha} (-1)^{i+1}. \hspace{1cm} (6-3)$$

Some properties of this expansion have been discussed in Appendix A.

In the following, we present our ensemble constructions for the two categories of code design.

### 6.2.1 Capacity achieving sequences for a given channel erasure probability

Consider a check node degree distribution $\rho(x)$, a given channel erasure probability $\varepsilon$, and a maximum variable node degree $N$. The following variable node degree distribution together with $\rho(x)$ form a capacity achieving ensemble as $N$ tends to infinity:

$$\lambda_i = \frac{T_i}{\varepsilon}, \quad 2 \leq i \leq f(N),$$

$$\lambda_i = 0, \quad f(N) + 1 \leq i \leq N - 1,$$

$$\lambda_N = 1 - \sum_{i=2}^{f(N)} \lambda_i. \hspace{1cm} (6-4)$$

In (6-4), $N$ must satisfy the following inequalities and $f$ is a strictly increasing function of $N$ and smaller than $N$.

$$\sum_{i=2}^{N} T_i > \varepsilon, \hspace{1cm} (6-5)$$

$$\varepsilon \geq \sum_{i=2}^{N-1} T_i. \hspace{1cm} (6-6)$$
We call such a sequence $GC$ which stands for Given Channel erasure probability\(^{15}\). Note that for brevity, we have omitted the notation ‘[..]‘ (around $f(N)$). It is also important to note that $f(N) = N - 1$ corresponds to a special case of $GC$ ensembles where all the $\lambda_i$ coefficients, $2 \leq i \leq N$, are nonzero.

**Theorem 6-1**: Consider a given check node degree distribution $\rho(x)$, and denote the $i$th term of the Taylor expansion of $\rho^{-1}(1 - x)$ at $x = 0$ by $T_i$, as in (6-1). For a given channel parameter $\varepsilon > T_2$ and a set of constituent variable node degrees from 2 to $N$ ($N > 2$), there always exists a unique $N$ that satisfies (6-5) and (6-6). For such $N$, in (6-4), $\lambda_N \geq 0$, and the convergence of the ensemble constructed based on (6-4) is ensured.

Note that if $\varepsilon$ does not satisfy the inequality $\varepsilon > T_2$, we need to decrease $T_2$ by increasing $d_e$.

Consider now an ensemble constructed based on $GC$ sequences for a given $\rho(x)$. The maximum variable node degree for this sequence is $N$ which satisfies (6-5) and (6-6). Based on part 1 of Lemma 5-1, if we decrease the maximum variable node degree from $N$ to $D_v < N$, with the same percentage of adjacent edges, the newly constructed ensemble has a higher rate but can be non-convergent. By choosing the smallest $D_v$ which results in a convergent ensemble, we can create a new ensemble, referred to as *Modified GC* or *MGC*. Note that the variable node degree distribution for this ensemble is the same as that of the $GC$ ensemble with $\lambda_{D_v}$ replacing $\lambda_N$.

---

\(^{15}\) Note that GC sequences correspond to Type-A and Type B ensembles for the first category of code design in Chapter 5.
6.2.2 Capacity achieving sequences for a given code rate

Consider a check node degree distribution \( \rho(x) \), and a given code rate \( R \). The following variable node degree distribution together with \( \rho(x) \) form a capacity achieving ensemble as \( N \), the maximum variable node degree, tends to infinity:

\[
\lambda_i = \frac{T_i}{\epsilon} = \frac{\sum_{i=2}^{f(N)} T_i (1/i - 1/N)}{d_v^{-1} - 1/N}, \quad 2 \leq i \leq f(N),
\]

\[
\lambda_i = 0, \quad f(N) + 1 \leq i \leq N - 1,
\]

\[
\lambda_N = 1 - \sum_{i=2}^{f(N)} \lambda_i.
\]

In (6-7), \( N \) must satisfy the following inequalities and \( f \) is a strictly increasing function of \( N \) and smaller than \( N \).

\[
\bar{d}_v^{-1} \sum_{i=2}^{N} T_i > \sum_{i=2}^{N} T_i / i
\]

\[
\bar{d}_v^{-1} \sum_{i=2}^{N-1} T_i \leq \sum_{i=2}^{N-1} T_i / i
\]

We refer to such ensembles as \( GR \) which stands for Given code Rate\(^{16}\). Note that the special case of \( f(N) = N-1 \), where all the coefficients \( \lambda_i, 2 \leq i \leq N \), are nonzero, corresponds to the sequences of [22].

*Theorem 6-2*: For a given code rate \( R \) and a given check node degree distribution (and thus a given \( \bar{d}_v^{-1} \)), if \( R < 1 - 2 / \bar{d}_v \), there always exists a unique value of \( N \) that satisfies

---

\(^{16}\) Note that GR sequences correspond to Type-A and Type B ensembles for the second category of code design in Chapter 5.
For such \( N \), in (6-7), \( \lambda N \geq 0 \), and the convergence of the ensemble constructed based on (6-7) is ensured.

Note that if the code rate \( R \) does not satisfy the inequality, we need to increase \( d_c \).

Similar to the case in Section 6.2.1, we can decrease the maximum variable node degree of \( GR \) sequences from \( N \) to a smaller value \( D_w \) and design a new \( GR \) ensemble for a given rate. The ensemble corresponding to the smallest \( D_w \) which is also convergent is called Modified \( GR \) or \( MGR \). Clearly an \( MGR \) ensemble has a larger threshold value compared to the corresponding \( GR \) ensemble.

It can be easily shown that for \( GC \) ensembles, the channel parameter \( \varepsilon \) and for \( GR \) ensembles the value of \( \varepsilon \) in (6-7) are the thresholds of the ensembles. A similar result has been proven in theorem 5-3.

6.3 Achieving the capacity, the choice of \( f(N) \), and optimality /quasi-optimality of the proposed sequences

In this section, we first show that \( GC \) and \( GR \) sequences are capacity achieving. For simplicity, we consider check-regular ensembles throughout this chapter. Before we can prove the main result, we prove the following lemma.

**Lemma 6-1:** For \( GC \) and \( GR \) sequences, if \( D_c \) tends to infinity, the value of \( N \) in Theorems 6-1 and 6-2 also tends to infinity. Moreover, \( N \) tends to infinity much faster than \( D_c \) does. This is such that \( \lim_{D_c \to \infty} \frac{D_c}{N^\beta} = \lim_{\alpha \to 0} \frac{1}{\alpha N^\beta} = 0, \forall \beta > 0 \alpha = 1/(D_c - 1) \).
**Theorem 6-3**: GC and GR ensemble sequences are capacity achieving as $D_c$ tends to infinity.

For a capacity achieving sequence, it is very important to see how fast it achieves the capacity as the check node degree increases. This property was primarily considered for the sequences discussed in [22]. A sequence $(\lambda_n, \rho_n)$ of degree distributions giving rise to codes of rate $R$ is called asymptotically quasi-optimal if a constant $\mu$ exists for which [22]:

$$\lim_{n \to \infty} \text{Sup}(\mu(\lambda_n, \rho_n)) = \mu$$

where for GR sequences:

$$\mu(\lambda_n, \rho_n) = \frac{\log(R)}{\log(3(\lambda_n, \rho_n))} \quad \text{and} \quad \Im(\lambda_n, \rho_n) = 1 - \frac{\varepsilon}{1 - R}$$

and constant $\mu$ is always greater than or equal to 1. The closer $\mu$ is to 1, the faster $\varepsilon$ approaches $1 - R$.

By extending this definition to GC sequences, a sequence $(\lambda_n, \rho_n)$ of degree distributions constructed for a given channel erasure probability $\varepsilon$ is called asymptotically quasi-optimal if a constant $\mu$ exists for which:

$$\lim_{n \to \infty} \text{Sup}(\mu(\lambda_n, \rho_n)) = \mu$$

where

$$\mu(\lambda_n, \rho_n) = \frac{\log(1 - \varepsilon)}{\log(3(\lambda_n, \rho_n))} \quad \text{and} \quad \Im(\lambda_n, \rho_n) = 1 - \frac{\varepsilon}{1 - R}.$$
\[
\lim_{n \to \infty} \sup (\Delta(\lambda_n, \rho_n)) = \Delta,
\]
where
\[
\Delta(\lambda_n, \rho_n) = \frac{\mathcal{I}(\lambda_n, \rho_n)}{R^{de}}.
\]
Again we can extend this definition to GC sequences. A sequence \((\lambda_n, \rho_n)\) of degree distributions constructed for a given channel erasure probability \(\varepsilon\) is called asymptotically-optimal if a constant \(\Delta\) exists for which:
\[
\lim_{n \to \infty} \sup (\Delta(\lambda_n, \rho_n)) = \Delta,
\]
where
\[
\Delta(\lambda_n, \rho_n) = \frac{\mathcal{I}(\lambda_n, \rho_n)}{(1-\varepsilon)^{de}}.
\]
It can be verified that an asymptotically optimal ensemble sequence is asymptotically quasi-optimal with \(\mu = 1\).

It is shown in [22] that check-regular GR ensembles with \(f(N) = N-1\) are asymptotically optimal. In the sequel, for check-regular sequences with arbitrary \(f(N)\), for each category of GC and GR sequences, we find conditions on \(f(N)\) which are sufficient for the sequences to be asymptotically quasi-optimal and optimal, respectively. To do this, we first propose the following important lemmas.

**Lemma 6-2:** For GC and GR ensemble sequences, the following relationship holds:
\[
\lim_{\alpha \to 0} \sup D^{-1}_e \log \left(1 - \frac{\varepsilon}{1-R}\right) = \lim_{\alpha \to 0} \frac{\alpha}{\alpha + 1} \log \left(1 - \frac{1}{R^{(N)^{a+1}}} - \frac{1}{\alpha N} \left(\frac{1}{(f(N))^a} - (1-\varepsilon)\right)\right)
\]

**Lemma 6-3:** For GC and GR ensemble sequences, the following relationships hold respectively:
\[
\lim_{a \to 0} \text{Sup}(1 - \varepsilon) D_{c}^{a} \left(1 - \frac{\varepsilon}{1 - R}\right) = \lim_{a \to 0} (1 - \varepsilon)^{\gamma(a)} \left(\frac{1}{f(N)^{a+1}} - \frac{1}{\alpha N \left(f(N)^{a} - (1 - \varepsilon)\right)}\right)
\]

\[
\lim_{a \to 0} \text{Sup} R D_{c}^{a} \left(1 - \frac{\varepsilon}{1 - R}\right) = \lim_{a \to 0} R^{\gamma(a)} \left(\frac{1}{f(N)^{a+1}} - \frac{1}{\alpha N \left(f(N)^{a} - R\right)}\right)
\]

**Lemma 6-4:** Consider GC and GR ensemble sequences for which the values of $N$ are computed based on the pair of equations (6-5), (6-6), and (6-8), (6-9), respectively. For the GC ensemble sequence with the channel parameter $\varepsilon$ we have

\[
\lim_{a \to 0} \frac{1}{N^{a}} = 1 - \varepsilon , \quad \text{and} \quad \lim_{a \to 0} \frac{(1 - \varepsilon)^{-1/a}}{N} = e^{\gamma} .
\]

where $\gamma = 0.577215665...$ is the Euler constant, and for the GR ensemble sequence with rate $R$, we have

\[
\lim_{a \to 0} \frac{1}{N^{a}} = R , \quad \text{and} \quad \lim_{a \to 0} \frac{R^{-1/a}}{N} = e^{\gamma} .
\]

**Lemma 6-5:** Consider check-regular GC and GR ensemble sequences for which the values of $N$ are computed based on the equation pairs (6-5), (6-6), and (6-8), (6-9), respectively. If function $f(N)$ is chosen such that

\[
\lim_{a \to 0} f(N)^{a} = M ,
\]

where $M$ is a constant greater than 1, and

\[
\lim_{a \to 0} \frac{f(N)}{N} = 0 ,
\]

we then have

\[
\lim_{a \to 0} \frac{f(N)}{N^a} = 0 .
\]
Theorem 6-4: GC and GR ensemble sequences for which the function \( f(N) \) is chosen such that \( \lim_{a \to 0} f(N)^a = M \), where \( M \) is a constant greater than 1 and \( \lim_{a \to 0} \frac{f(N)}{N} = 0 \), are asymptotically quasi-optimal with constants \( \mu = -\log R / \log M \) and 
\( \mu = -\log(1-e)/\log M \), respectively.

Corollary 6-1: GC and GR sequences with \( f(N) = aN^\beta + b \), where \( 0 < a \leq 1 \) and \( 0 < \beta \leq 1 \) and \( b \) is an arbitrary constant, are asymptotically quasi-optimal with \( \mu = 1/\beta \).

Note that the case where \( \lim_{a \to 0} \frac{f(N)}{N} = K > 0 \), which is equivalent to \( M \) being equal to 
\( 1/(1-e) \) and \( 1/R \) for GC and GR sequences respectively, will be proven in the next theorem to be asymptotically optimal which implies their quasi-optimality as well.

Theorem 6-5: GC and GR sequences are asymptotically optimal with
\( \Delta = Q - (1-e)^{-1}e^\gamma \) and \( \Delta = Q - R^{-1}e^\gamma \), respectively, if

for GC ensembles, \( \lim_{a \to 0} \frac{(1-e)^{-1/a}}{f(N)} = Q \) and \( \lim_{a \to 0} \frac{1}{a} \left( \frac{1}{f(N)^a} - (1-e) \right) = \gamma \), and

for GR ensembles, \( \lim_{a \to 0} \frac{R^{-1/a}}{f(N)} = Q \) and \( \lim_{a \to 0} \frac{1}{a} \left( \frac{1}{f(N)^a} - R \right) = \gamma \),

where \( Q \) and \( \gamma \) are positive constants independent of \( N \) and \( \alpha \) and \( \gamma \) is the Euler constant.
Corollary 6-2: GC and GR sequences with \( f(N) = aN^\beta + b \), where \( 0 < \beta, a \leq 1 \), and \( b \) is an arbitrary constant, are asymptotically optimal with \( \Delta = (\frac{1}{a} + \ln a)e^\gamma \) if \( \beta = 1 \), i.e., if \( f(N) \) is a linear function of \( N \).

Note that for an asymptotically optimal sequence, the smaller the \( \Delta \), the faster it achieves the capacity. For GR and GC sequences with \( f(N) = aN + b \), the smallest value for \( \Delta \) happens when \( a = 1 \). This corresponds to the check-regular sequences proposed in [21][22] with \( \Delta = e^\gamma \). For other values of \( a \), \( \Delta \) is larger. We note that since MGC and MGR sequences always perform better than GC and GR sequences, respectively, they are also capacity achieving following the result proved in Lemma 5-1. Results similar to those of Theorems 6-2 and 6-3 also applies to MGC and MGR sequences, although we have not been able to compute the constants \( \mu \) and \( \Delta \) for these sequences analytically. In the next section, we show that for MGC and MGR sequences with \( f(N) = aN + b \), for a given value of \( a \), the value of \( \Delta \) tends to a value between \( e^\gamma \) and \( e^\gamma (1/a + \ln a) \) and in fact very close to \( e^\gamma \).

6.4 Simulation results

In this section, we consider GR ensembles with rate one half. For other code rates and also for GC ensembles similar results are obtained. We consider check-regular sequences with different \( f(N) \) functions and compute the values of \( \mu \), \( \Delta \) and \( \Delta \). Consider GR ensemble sequences \( C_1 \), \( C_2 \), \( C_3 \), and \( C_4 \) with \( f_1(N) = [\ln(N)] + 2 \), \( f_2(N) = [\sqrt{N}] + 2 \), \( f_3(N) = [N/4] + 2 \) and \( f_4(N) = N - 1 \) respectively. Also let \( C_4 \) be
MGR ensemble with \( f_4(N) = \lceil N/4 \rceil + 2 \). Note that \( C_3 \) is the same as the rate one half check regular ensemble proposed in [22]. The constant values have been used to ensure that for smaller values of check node degrees (and thus smaller values of \( N \)), we have at least 3 constituent variable node degrees. Note that for large values of \( N \), the effect of the constant is negligible. The results have been reported in Table 6-1.

As can be seen in Table 6-1, while all ensembles are capacity achieving, their speed of achieving the capacity is different. Ensemble \( C_1 \) achieves the capacity very slowly. It can be seen that \( \mu_1 \) increases as \( D_c \) increases and the ensemble is not asymptotically quasi-optimal. Ensemble \( C_2 \) is asymptotically quasi-optimal and the value of \( \mu_2 \) tends to 2 based on Corollary 6-1. It can be seen that \( \mu_2 \) does not decrease uniformly to its limit as opposed to \( \mu_3, \mu_4 \) and \( \mu_5 \). Ensembles \( C_3 \) and \( C_4 \) are asymptotically optimal. Based on Corollary 6-2, the value of \( \Lambda_3 \) tends to \((4 - \log 4)e^r = 4.65\). By comparing the values of \( \mu \) and \( \Lambda \) for \( C_3 \) and \( C_4 \), it can also be seen that \( C_4 \) converges faster to the capacity compared to \( C_3 \), as expected. This is in addition to the advantage of \( C_4 \) over \( C_3 \) in having a smaller maximum variable node degree.

Another important comparison is to see that the values of \( \Delta_4 \) are very close to those of \( \Delta_5 \) for a given \( D_c \). This means that the MGR ensembles introduced here have more or less the same speed of convergence to the capacity as the check-regular ensembles of [22] do. This is while \( C_4 \) has a much smaller number of constituent variable node degrees (by a factor of about 4 for larger \( D_c \) values) and a much smaller maximum variable node degree compared to \( C_5 \).
In general, the results of the table suggest that choosing linear functions of $N$ for $f(N)$ results in ensembles that perform very closely to those proposed in [22] but with a considerably smaller number of constituent variable node degrees and a smaller maximum variable node degree. We therefore concentrate on $f_a(N) = [aN] + b$ and study the effect of $a$ on the performance of the proposed ensembles. Note again that for $a = 1$ and $b = -1$, the resulting ensemble sequence is identical to that of [22].

### Table 6-1: The values of $\mathcal{I}$, $\mu$ and $\Delta$ for different ensemble sequences and different check node degrees.

<table>
<thead>
<tr>
<th>$D_c$</th>
<th>$N$</th>
<th>$\mathcal{I}_1$</th>
<th>$\mathcal{I}_2$</th>
<th>$\mathcal{I}_3$</th>
<th>$\mathcal{I}_4$</th>
<th>$\mathcal{I}_5$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\mu_3$</th>
<th>$\mu_4$</th>
<th>$\Delta_1$</th>
<th>$\Delta_2$</th>
<th>$\Delta_3$</th>
<th>$\Delta_4$</th>
<th>$\Delta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>.1127</td>
<td>.2857</td>
<td>.127</td>
<td>.1128</td>
<td>.1020</td>
<td>1.587</td>
<td>2.765</td>
<td>1.587</td>
<td>1.587</td>
<td>1.518</td>
<td>3.60</td>
<td>3.60</td>
<td>3.60</td>
<td>3.60</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>.0829</td>
<td>.1523</td>
<td>.0829</td>
<td>.0362</td>
<td>.0494</td>
<td>1.675</td>
<td>1.924</td>
<td>1.67</td>
<td>1.723</td>
<td>1.296</td>
<td>5.30</td>
<td>5.30</td>
<td>5.30</td>
<td>5.30</td>
</tr>
<tr>
<td>7</td>
<td>29</td>
<td>.1047</td>
<td>.1047</td>
<td>.0552</td>
<td>.0206</td>
<td>.0179</td>
<td>2.153</td>
<td>2.150</td>
<td>1.516</td>
<td>1.248</td>
<td>1.206</td>
<td>5.40</td>
<td>5.40</td>
<td>5.40</td>
<td>5.40</td>
</tr>
<tr>
<td>8</td>
<td>61</td>
<td>.0972</td>
<td>.0657</td>
<td>.0233</td>
<td>.0084</td>
<td>.0084</td>
<td>2.378</td>
<td>2.110</td>
<td>1.474</td>
<td>1.161</td>
<td>24.8</td>
<td>25.4</td>
<td>6.16</td>
<td>5.95</td>
<td>2.16</td>
</tr>
<tr>
<td>9</td>
<td>126</td>
<td>.0909</td>
<td>.0509</td>
<td>.0113</td>
<td>.0042</td>
<td>.0040</td>
<td>2.601</td>
<td>2.094</td>
<td>1.390</td>
<td>1.138</td>
<td>13.0</td>
<td>20.6</td>
<td>5.76</td>
<td>2.13</td>
<td>2.05</td>
</tr>
<tr>
<td>10</td>
<td>257</td>
<td>.0850</td>
<td>.0365</td>
<td>.0057</td>
<td>.0020</td>
<td>.0019</td>
<td>2.812</td>
<td>2.090</td>
<td>1.342</td>
<td>1.126</td>
<td>11.10</td>
<td>37.1</td>
<td>5.85</td>
<td>2.10</td>
<td>1.99</td>
</tr>
<tr>
<td>11</td>
<td>523</td>
<td>.0911</td>
<td>.0262</td>
<td>.0010</td>
<td>.0009</td>
<td>3.181</td>
<td>2.093</td>
<td>1.296</td>
<td>1.105</td>
<td>1.096</td>
<td>186</td>
<td>53.6</td>
<td>5.70</td>
<td>2.07</td>
<td>1.96</td>
</tr>
<tr>
<td>12</td>
<td>1059</td>
<td>.0836</td>
<td>.0188</td>
<td>.0014</td>
<td>.0004</td>
<td>.0005</td>
<td>3.351</td>
<td>2.093</td>
<td>1.261</td>
<td>1.093</td>
<td>1.086</td>
<td>342</td>
<td>77.0</td>
<td>5.61</td>
<td>2.04</td>
</tr>
<tr>
<td>13</td>
<td>2136</td>
<td>.0771</td>
<td>.0139</td>
<td>.0007</td>
<td>.0002</td>
<td>.0002</td>
<td>3.515</td>
<td>2.106</td>
<td>1.234</td>
<td>1.065</td>
<td>1.078</td>
<td>631</td>
<td>113</td>
<td>5.54</td>
<td>2.02</td>
</tr>
<tr>
<td>14</td>
<td>4301</td>
<td>.0792</td>
<td>.0099</td>
<td>.0003</td>
<td>.0001</td>
<td>.0001</td>
<td>3.811</td>
<td>2.100</td>
<td>1.212</td>
<td>1.077</td>
<td>1.071</td>
<td>1297</td>
<td>161</td>
<td>5.47</td>
<td>2.00</td>
</tr>
<tr>
<td>15</td>
<td>8650</td>
<td>.0730</td>
<td>.0071</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>4.001</td>
<td>2.102</td>
<td>1.193</td>
<td>1.071</td>
<td>1.065</td>
<td>2391</td>
<td>323</td>
<td>5.40</td>
<td>1.99</td>
</tr>
<tr>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

In the remainder of this section, we consider check-regular ensembles $C_4, C_6, C_7, C_8,$ and $C_9$ corresponding to $a = 1/4, 1/2, 1/3, 1/6, and 1/8$ respectively and $b = 2$. For each value of $a$ and each check node degree, we construct an MGR ensemble for which $N$ is obtained from (6-8) and (6-9). For this ensemble, the number of constituent variable node degrees $P$ is equal to $f_a(N)$, and we use the ratio of the ensemble threshold $\psi_a$ to the best achievable threshold $1 - R = 1/2$ as the measure of performance ($\psi = \psi_a / (1 - R)$). The results are reported in Table 6-2. The results of Table 6-2 also show that, as expected, for a given $D_c$, the larger the $P$, the better the performance, and the best performance is achieved for $a = 1$ (some inconsistencies for small values of $D_c$ might be caused by the
constant term \( b \). However, a similar performance can be achieved by the proposed MGR ensembles with considerably smaller \( P \) and \( D_v \). This is achieved at the expense of a slight increase in \( D_c \) (in most cases, \( D_c \) has to increase by only one and in very few cases by two).

Table 6-2: The values of \( P, D_v \) and \( \psi \) for different ensemble sequences with \( f(N) = aN + b \) and different check node degrees.

<table>
<thead>
<tr>
<th>( D_c )</th>
<th>( N )</th>
<th>( w_5 )</th>
<th>( P_s )</th>
<th>( D_{in} )</th>
<th>( w_3 )</th>
<th>( P_s )</th>
<th>( D_{in} )</th>
<th>( w_5 )</th>
<th>( P_s )</th>
<th>( D_{in} )</th>
<th>( w_5 )</th>
<th>( P_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>.8980</td>
<td>5</td>
<td>7</td>
<td>.8956</td>
<td>4</td>
<td>6</td>
<td>.8873</td>
<td>4</td>
<td>6</td>
<td>.8872</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>.9596</td>
<td>9</td>
<td>11</td>
<td>.9596</td>
<td>6</td>
<td>9</td>
<td>.9596</td>
<td>5</td>
<td>8</td>
<td>.9596</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>29</td>
<td>.9821</td>
<td>17</td>
<td>22</td>
<td>.9821</td>
<td>12</td>
<td>18</td>
<td>.9821</td>
<td>9</td>
<td>16</td>
<td>.9794</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>61</td>
<td>.9916</td>
<td>33</td>
<td>43</td>
<td>.9916</td>
<td>22</td>
<td>12</td>
<td>.9916</td>
<td>17</td>
<td>30</td>
<td>.9916</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>126</td>
<td>.9990</td>
<td>65</td>
<td>86</td>
<td>.9960</td>
<td>44</td>
<td>70</td>
<td>.9960</td>
<td>34</td>
<td>61</td>
<td>.9958</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>257</td>
<td>.9991</td>
<td>131</td>
<td>173</td>
<td>.9991</td>
<td>88</td>
<td>88</td>
<td>.9981</td>
<td>66</td>
<td>121</td>
<td>.9980</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>523</td>
<td>.9991</td>
<td>264</td>
<td>350</td>
<td>.9991</td>
<td>176</td>
<td>282</td>
<td>.9991</td>
<td>133</td>
<td>245</td>
<td>.9990</td>
<td>89</td>
</tr>
<tr>
<td>12</td>
<td>1059</td>
<td>.9995</td>
<td>532</td>
<td>705</td>
<td>.9995</td>
<td>355</td>
<td>569</td>
<td>.9995</td>
<td>267</td>
<td>493</td>
<td>.9996</td>
<td>179</td>
</tr>
<tr>
<td>13</td>
<td>2136</td>
<td>.9998</td>
<td>1070</td>
<td>1440</td>
<td>.9998</td>
<td>714</td>
<td>1145</td>
<td>.9998</td>
<td>536</td>
<td>993</td>
<td>.9998</td>
<td>358</td>
</tr>
<tr>
<td>14</td>
<td>4301</td>
<td>.9999</td>
<td>2153</td>
<td>2858</td>
<td>.9999</td>
<td>1436</td>
<td>2305</td>
<td>.9999</td>
<td>1077</td>
<td>1997</td>
<td>.9999</td>
<td>719</td>
</tr>
</tbody>
</table>

In Figure 6-1 and Figure 6-2, we have shown the values for \( P \) and \( D_v \) versus the performance of the ensembles for different values of \( a \) and for practical values of \( D_c \) (\( D_c \) values are chosen such that the corresponding values of \( D_v \) are less than 200). As can be seen in Figure 6-1 except for \( a = 1/8 \) all the other values of \( a \) require a smaller \( D_v \) compared to the ensemble with \( a = 1 \) to achieve a certain performance. It is also seen in Figure 6-2 that for all values of \( a \neq 1 \), the number of required variable nodes is less than that of \( a = 1 \) for the same performance. Again note that in the figures, for a given performance, the \( D_c \) values for different ensembles are only slightly different (by one or two). The figures also show that for \( R = 1/2 \) and for the tested range of \( D_c \) values, \( a = 1/4 \) is the overall best choice in terms of both requiring the lowest number of constituent variable nodes and smallest maximum variable node degrees for a given performance.
6.5 Concluding remarks

In this chapter we presented new set of capacity achieving sequences of LDPC code ensembles. We showed that similar to the sequences proposed by Shokrollahi, for any given \( \delta, 0 < \delta < 1-R \), they can achieve a threshold within \( \delta \)-neighborhood of the capacity upper bound \( 1-R \). We also showed that for such \( \delta \), some of our sequences in the set are superior to Shokrollahi’s sequences in that for each of them, there exists an integer number \( n_0 \), such that for any \( n > n_0 \), the sequence \( (\lambda^n, \rho^n) \) requires a smaller maximum variable node degree as well as a smaller number of constituent variable node degrees. We also showed that with proper choice of \( f(N) \), such sequences are asymptotically quasi-optimal. We also proved that by choosing \( f(N) \) as a linear function of \( N \), our proposed sequences are asymptotically optimal.
Figure 6-1: The required maximum variable node degree for different performances and different values of $a$.

Figure 6-2: The required number of constituent variable node degree for different performances and different values of $a$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Chapter 7 A deterministic approach for designing LDPC codes over BIAWGN channels

7.1 Introduction

In this chapter we extend our findings over the BEC to the AWGN channel. To be able to deterministically design LDPC codes over a BIAWGN the key point is to find similar upper bounds to those for BEC. After finding such upper bounds, we design LDPC codes for both categories of code design and show that their performance is close
to optimal. We note however that as opposed to Chapter 5, the methods proposed in this section can not be considered completely deterministic. We use the same basic ideas of Chapter 5 but after the initial ensemble is designed deterministically, some ‘fine tuning’ might be necessary to obtain the best result. The complexity of the proposed algorithms however is still considerably less than that of the optimization-based methods. Throughout this chapter, we use density evolution [7] to check if an ensemble converges for a given channel parameter.

7.2 Upper bounds on variable node degrees over the BIAWGN channel

It is well-known that over the BEC and BIAWGN channel, for a given check node degree distribution and a given channel parameter, there is an upper bound $\lambda_{2,\text{max}}$ on $\lambda_2$ such that if $\lambda_2 > \lambda_{2,\text{max}}$, the probability of error is bounded away from zero regardless of the value of the other $\lambda_i$'s and the number of iterations [7]. This is also known as the stability condition.

From Chapter 5, for degree 2 variable nodes over the BEC we have the following upper bound:

$$\lambda_2 \leq T_2 / \varepsilon,$$

where $T_i$ is the $i$th Taylor series coefficient of $1 - \rho^{-1}(1 - x)$. For other $\lambda_i$'s, $n > 2$, it was shown in Chapter 5 that an upper bound exists only if $\lambda_i = T_i / \varepsilon, 2 \leq i \leq n - 1$. This means that starting from degree 3, if the value of $\lambda_2$ is set equal to its upper bound, there is an upper bound on $\lambda_3$ such that if $\lambda_2$ is set greater than this upper bound, the probability of error is bounded away from zero regardless of the values of other $\lambda_i$'s and the number of
iterations. Similarly, if $\lambda_2$ and $\lambda_3$ have their values equal to their upper bounds, there would be an upper bound for $\lambda_4$ and so on. The values of these upper bounds are equal to $T_n / \epsilon$ for the BEC.

For the BIAWGN channel with noise variance $\sigma^2$, we have [7]:

$$
\lambda_2 \leq e^{1/2\sigma^2} / \rho'(1).
$$

(7-1)

The value of the upper bound can be seen as a function of the check node degree distribution and the channel noise power. In other words:

$$
\lambda_{2,\text{max}} = M_2(\rho(x), \rho'(x), \rho''(x), \ldots |_{x=1}, \sigma).
$$

Also note that the upper bound on $\lambda_2$ is a decreasing function of $\sigma^2$ and $\rho'(1)$. The existence of the upper bound for degrees greater than 2 for BEC encourages us to see whether such upper bounds exist for BIAWGN channel as well. Starting from degree 3 by setting $\lambda_2 = \lambda_{2,\text{max}}$ and based on numerical experiments, we would like to see whether an upper bound $\lambda_{3,\text{max}}$ exists such that if $\lambda_3 > \lambda_{3,\text{max}}$, the probability of error is bounded away from zero regardless of the value of the other $\lambda_i$’s and number of iterations. Moreover if such upper bound exists, we would like to find a mapping $M_3$ similar to $M_2$, possibly a function of the check node degree distribution and channel noise power, such that

$$
\lambda_{3,\text{max}} = M_3(\rho(x), \rho'(x), \rho''(x), \ldots |_{x=1}, \sigma).
$$

To simplify our task, in our experiments we consider check node degree distributions in the form of

$$
\rho(x) = \alpha x^{Dc-2} + (1-\alpha)x^{Dc-1}.
$$

(7-2)
Note that for any value of average check node degree \( \bar{d}_c \), there are unique values for \( D_c \) and \( \alpha \) as follows:

\[
D_c = \left\lfloor \bar{d}_c \right\rfloor + 1
\]

\[
\alpha = (D_c - 1)(D_c / \bar{d}_c - 1).
\]

where \( \left\lfloor \cdot \right\rfloor \) means the smallest closest integer. In other words, degree distribution (7-2) can fit any given \( \bar{d}_c \). In fact structure (7-2) is commonly used to design LDPC code ensembles see e.g., [7][8][36]. Throughout this chapter, we assume that all check node degree distributions have the structure (7-2).

In the sequel, for different values of \( \bar{d}_c \) and channel noise power \( \sigma^2 \), we compute the upper bound on \( \lambda_3 \) using the following algorithm.

**Algorithm 7-1**

1- Set \( \lambda_{2,\text{max}} = e^{\frac{1}{2} \sigma^2} / \rho'(1) = \frac{1}{(D_c - 1)(1 - \alpha) + (D_c - 2)\alpha} e^{\frac{1}{2} \sigma^2} \).

2- Set \( D_v \) to a large enough value (this will be discussed later) and set \( \lambda_3 = 0 \).

3- Check if \( \lambda_{2,\text{max}} < 1 \). If not, then stop the algorithm as there is no upper bound on \( \lambda_3 \), otherwise go to the next step.

4- Increase \( \lambda_3 \) by the step size\(^{17} \) and construct a variable node degree distribution as follows:

\[
\lambda(x) = \lambda_{2,\text{max}} x + \lambda_3 x^2 + (1 - \lambda_3 - \lambda_{2,\text{max}}) x^{D_v - 1}.
\]

\(^{17}\) The value of step size depends on the required accuracy. We have used a step size of .001 which is small enough for our purpose in this work.
5-Check if \( 1 - \lambda_3 - \lambda_{2,\text{max}} \geq 0 \). If not, then stop the algorithm as there is no upper bound on \( \lambda_3 \). Otherwise go to the next step.

6-Check if the probability of error goes to zero after a large number of iterations. If yes, go to Step 4. Otherwise go to the next step.

7-The last value for \( \lambda_3 \) would be the upper bound on \( \lambda_3 \).

The value of \( D_v \) in Step 2 has to be chosen large enough such that the density evolution does not get trapped in any point other than the fixed point of zero. Therefore, for a \( \lambda_{3,\text{max}} \) computed by Algorithm 7.1 and based on a given \( D_v \), if we increase \( D_v \), the probability of error still has to be bounded away from zero for any \( \lambda_3 > \lambda_{3,\text{max}} \). For check node degree distributions with average degree less than 9, we observe that \( D_v = 30 \) is sufficiently large.

**Discussion 7.1:** In order to be able to set \( \lambda_2 = e^{1/2\sigma^2} / \rho'(1) \), as in Step 1 of the algorithm, the following inequality must hold

\[
e^{1/2\sigma^2} / \rho'(1) \leq 1.
\]  \hspace{1cm} (7-3)

This inequality is similar to the one for the BEC stated in Theorem 5.1 and is necessary to make sure that \( \lambda_2 \leq 1 \) which guarantees \( \lambda_{D_v} \geq 0 \). If (7-3) is not satisfied, no upper bound on \( \lambda_3 \) can be defined. This condition is checked in Step 3 of the algorithm.

When (7-3) is satisfied and we set \( \lambda_2 = e^{1/2\sigma^2} / \rho'(1) \), a similar situation might happen for \( \lambda_1 \), i.e., for a certain value of \( \lambda_3 \), the ensemble converges, and then for the new value of \( \lambda_3 \) in the next iteration of the algorithm, we have \( \lambda_{D_v} = 1 - \lambda_3 - \lambda_{2,\text{max}} < 0 \). This situation
has been taken care of in Step 5 and if it happens, again no upper bound on $\lambda_3$ can be defined. The following example makes the above discussion more clear.

**Example 7-1:** Consider a BIAWGN channel with the channel noise power $\sigma^2 = 0.4$ and the check node degree distribution $\rho(x) = x^3$. We have $e^{\frac{1}{2}\sigma^2} \rho'(1) = 1.1634 > 1$, which violates (7-3). Now if we increase the check node degree from 4 to 5 and the channel noise variance from 0.4 to 0.5, we have $e^{\frac{1}{2}\sigma^2} / \rho'(1) = 0.6796 < 1$ which satisfies (7-2)(7-3). This requires that $\lambda_3 \leq 1 - 0.6796 = 0.3204$. Therefore the value of $\lambda_{3,\text{max}}$ (if an upper bound actually exists) has to be less than 0.3204. In other words, for an upper bound on $\lambda_3$ to exist, while performing Algorithm 7-1 and increasing $\lambda_3$, there has to be a value of $\lambda_3 < 0.3204$ for which the constructed ensemble does not converge. Performing the algorithm for this example we observe that this is not the case. For the given check node degree, the smallest value of $\sigma^2$ for which the upper bound on $\lambda_3$ exists is 0.6970. For this value of $\sigma^2$ we have $\lambda_{2,\text{max}} = 0.5123$ and $\lambda_{3,\text{max}} = 0.4877$. Note that in this case $\lambda_{2,\text{max}} + \lambda_{3,\text{max}} = 1$.

After performing the Algorithm 7-1 for different $\sigma^2$ and $\bar{d}_e$ in the range of interest, we store the obtained values in a lookup table to be used later. These values are also plotted in Figure 7-1. As can be seen, similar to $\lambda_{2,\text{max}}$, $\lambda_{3,\text{max}}$ is a decreasing function of $\sigma^2$ and $\bar{d}_e$.

Unfortunately we have not been able to obtain upper bounds for other variable node degrees so far. However based on our experiences with the rate one half codes,
good ensembles can be designed using degrees 2, 3 and a maximum variable node degree less than 10. For higher values of $D_v$ setting $\lambda_2$ and $\lambda_3$ to their maximum values and using one extra constituent variable node degree (in addition to $D_v$) in an optimization loop would again result in close-to-optimum ensembles.

Figure 7-1: The values of $\lambda_{3,\text{max}}$ versus $\overline{d}_c$ and $\sigma^2$ the noise variance for BIAWGN channel.
7.3 Deterministic design of LDPC codes for a given channel parameter

In this section we propose two algorithms to design variable node degree distributions for check node degree distribution with the structure in (7-2) and a given channel parameter. In the first algorithm, we assume that the check node degree distribution is given. In this case we look for the smallest maximum variable node degree $D_v$ for which the constructed ensemble is convergent. In other words, the first algorithm is not able to design an ensemble with a constraint on $D_v$. For this reason we also propose a second algorithm to design codes with a given maximum variable node degree. In this case the check node degree distribution has to be properly designed based on the given $D_v$. These results will then be extended to the case of fixed rate ensembles in the next section. In order for both algorithms to produce a satisfactory result, the check node degree distribution and the channel noise variance should satisfy the conditions mentioned in Discussion 7-1.

Algorithm 7-2:

Consider the channel noise variance $\sigma^2$ and the check node degree distribution $\rho(x)$ as the input to the algorithm.

2. Set $\lambda_i = \lambda_{i,\text{max}} - 0.001, \ i = 2,3$.
3. Set $\lambda_{D_v} = 1 - \sum_{i=2}^{3} \lambda_i$.
4. Check to see if the constructed ensemble is convergent. If it is not, then increase $D_v$ by one and go to Step 3. Otherwise, the current $(\lambda(x), \rho(x))$ is the desired ensemble.
Note that is Step 2, we have subtracted the small value of .001 from the upper bounds of $\lambda_2$ and $\lambda_3$. Based on our experience this is desirable to obtain a faster convergence.

As a point of reference, we consider an ensemble from Table I of [7] with the following degree distributions:

$$\lambda(x) = .30013x + .28395x^2 + .41592x^7,$$
$$\rho(x) = .22919x^5 + .77081x^6.$$  

This ensemble which we call $C$, has rate one half and SNR threshold of 0.4483 dB. The maximum variable node degree of this code is 8. In fact, with the given check node degree distribution and the constituent variable node degrees, this is the best rate one half code that can be designed. This implies that within the given structure for variable and check node degrees, for the channel signal to noise ratio of .4483 dB (or equivalently channel noise variance of 0.9019), the highest achievable rate is 0.5.

**Example 7-2:** We would like to design a code for the channel noise variance $\sigma^2 = 0.9019$, and the following check node degree distribution:

$$\rho(x) = .22919x^5 + .77081x^6.$$  

We have:

$$\lambda_2 = \lambda_{2,\text{max}} - .001 = e^{1/2(0.9019)}/(5 \times .22919 + 6 \times .77081) - .001 = .3007,$$
$$\lambda_3 = \lambda_{3,\text{max}} - .001 = 0.2825 \text{ and } \lambda_{D_v} = 1 - \lambda_2 - \lambda_3 = .4168.$$  

The smallest value of $D_v$ that results in a convergent ensemble is 9 and thus we have

$$\lambda(x) = .3007x + .2825x^2 + .4168x^8.$$
The code rate of this ensemble is 0.4900 which is reasonably close to 0.5. The main reason for the gap is that the maximum variable node degree has increased from 8 to 9. The increase in $D_v$ is necessary for the code to be convergent. By modifying the check node degree distribution we obtain the following degree distribution pair with $D_v = 8$:

$$\lambda(x) = 0.3022x + 0.2827x^2 + 0.4151x^7,$$

$$\rho(x) = 0.258x^5 + 0.742x^6.$$

The rate of this ensemble is 0.4987 which is much closer to 0.5.

As shown in the above example, slight modification of the check node degree distribution can be very effective in obtaining an ensemble whose performance is close to optimum. The main reason is that while $D_v$ is an integer, $\bar{d}_c$ and $\rho(x)$ can be finely adjusted in smaller steps to maximize the rate. Besides, we might be interested in designing an ensemble with a given $D_v$. The following algorithm makes it possible to achieve these goals systematically.

**Algorithm 7-3:**

Assume that the channel noise variance $\sigma^2$ and the maximum variable node degree $D_v$ are given. Also assume the following structure for the degree distribution:

$$\rho(x) = \alpha x^{D_v-2} + (1 - \alpha)x^{D_v-1}.$$

1- Set $D_c$ to a large value (for the considered cases, 9 is large enough) and $\alpha$ to 0.

2- Set $\lambda_i = \lambda_{i,\text{max}} - 0.001$, $i = 2, 3$ and $\lambda_{D_v} = 1 - \sum_{i=2}^{3} \lambda_i$.

3- Check if the constructed ensemble is convergent. If it is, go to Step 4, otherwise decrease the average check node degree. This is done first by increasing $\alpha$ at small
steps ¹⁸ as long as $\alpha < 1$. Then if $\alpha = 1$, $D_c$ is decreased by one and $\alpha$ is reset to 0.

Go to Step 2.

4- The current $(\lambda(x), \rho(x))$ is the desired ensemble.

Similar to Algorithm 7-2, the check node degree distribution and the channel noise variance should satisfy conditions mentioned in Discussion 7-1. If in the process of modifying the check node degree distribution, we reach to a point where no upper bound can be found for $\lambda_3$, the algorithm stops and no ensemble with the given $D_v$ can be designed. Also note that to obtain the best result, we have to use a large enough value for $D_c$ in the beginning of the algorithm. This can be verified by making sure that when Step 3 of the algorithm is performed for the first time, the designed code is non-convergent.

Although Algorithms 7-2 and 7-3 are not fully deterministic, they are much less complex than optimization-based design methods. Note that the most computationally demanding part of both algorithms is the convergence test which is performed by density evolution. In Algorithm 7-2, one needs to perform the convergence test for each new maximum variable node degree. For instance in Example 7-2, we started with $D_v = 4$ and had to perform the convergence test before we reached to the final result of $D_v = 9$. For Algorithm 7-3 one needs to perform one convergence test for each tested check node degree distribution. In an optimization-based design however, the convergence test has to be performed for different combinations of check and variable degree distributions. The number of such combinations is usually much larger than the number of tested combinations for only the check node distribution, as performed in Algorithm 7-3.

¹⁸ The step size is determined based on the required precision. In this work we use step size of 0.001.
7.4 Deterministic design of LDPC codes for a given code rate

We first note that to design ensembles with a given rate $R$ and a given check node degree distribution $\rho(x)$ with an average check node degree $\overline{d}_c$, the following inequality must hold

$$R \leq 1 - \frac{2}{\overline{d}_c}. \quad (7-4)$$

Our approach is to design an ensemble for a given channel parameter and then change that channel parameter to obtain the desired rate in an iterative fashion. Suppose that for a given check node degree distribution and an arbitrary channel noise variance $\sigma^2$, we design an ensemble by performing the first 3 steps of Algorithm 7-2. At this stage, the constructed ensemble may or may not be convergent. The code rate of this ensemble can be computed by:

$$R = 1 - \left[ \frac{\lambda_{2,\text{max}}(\sigma^2)}{2} + \frac{\lambda_{3,\text{max}}(\sigma^2)}{3} + \frac{\lambda_{D_2}(\sigma^2)}{D_v} \right]^{-1} \overline{d}_c^{-1} = f(\sigma^2), \quad (7-5)$$

where $f$ is a (continuous) function of $\sigma^2$. Knowing that $\lambda_{2,\text{max}}$ and $\lambda_{3,\text{max}}$ are decreasing functions of $\sigma^2$, it can be seen that $f$ is a strictly decreasing function of $\sigma^2$. In fact, it can be verified that for a given $0 < R < 1$, equation (7-5) has a unique solution for $\sigma^2$. Therefore for the desired ensemble, if $R$ is smaller (larger) than the target rate we can close the gap by decreasing (increasing) $\sigma^2$. Moreover, to ensure the convergence of the designed ensemble, we need to add a step similar to Step 4 of Algorithm 7-2. The following algorithm is the parallel of Algorithm 7-2 for the case where the code rate is given.

*Algorithm 7-4:*
Assume that the code rate $R^*$ and the check node degree distribution $\rho(x)$ as the input to the algorithm.

1- Set $D_v = 4$, and compute the largest possible channel noise power $\sigma^2_{\text{BIAWGN}}$ for reliable transmission with the given rate $R^*$ using the Shannon capacity of the BIAWGN channel.

2- Set the channel noise parameter $\sigma^2 = \sigma^2_{\text{BIAWGN}}$.

3- Deterministically construct an ensemble by setting

$$\lambda_i = \lambda_{i,\text{max}} - 0.001, \quad i = 2, 3 \text{ and } \lambda_{D_v} = 1 - \sum_{i=2}^{3} \lambda_i.$$ 

4- Compute $R$, the rate of the resulting ensemble, and compare it with $R^*$.

5- If $|R - R^*| < q$, where $q$ is the desired precision then go to the next step. Otherwise, if $R < R^*$, decrease $\sigma^2$ and if $R > R^*$, increase $\sigma^2$. The step size\(^{19}\) in changing $\sigma^2$ depends on $q$. In both cases go to Step 3.

6- Check if the resulting ensemble is convergent for the current value of $\sigma^2$. If no, increase $D_v$ by one and go to Step 2. If yes, the current $(\lambda(x), \rho(x))$ is the desired ensemble. The threshold of the ensemble is the current value of $\sigma^2$.

Note that as the algorithm proceeds and before a $\sigma^2$ is found for which $|R - R^*| < q$, only steps 3 to 5 are performed. Once $|R - R^*| < q$, we need to perform the convergence test. Since the convergence test is only performed once for each value of $D_v$, and the computational complexity of steps 3 to 5 are negligible compared to that of the

\(^{19}\) For $q=.00001$, based on our experience, the step size of .00005 is sufficiently small.
convergence test, the complexity of this algorithm is almost the same as that of Algorithm 7-2.

Example 7-3: Consider the design of a rate one half code with check node degree distribution \( \rho(x) = 0.22919x^5 + 0.77081x^6 \). This distribution is the same as that of ensemble \( C \) defined in Section 7.2. After performing Algorithm 7-4, we obtain the following variable node degree distribution

\[
\lambda(x) = 0.3089x + 0.2940x^2 + 0.3971x^8
\]

The threshold for this code is 0.6544 dB which is more than 0.2 dB away from that of \( C \). Again this gap is due to the increase of \( D_v \) from 8 to 9. Similar to Example 7-2, by slightly modifying the check node degree distribution, we are able to obtain the following ensemble with \( D_v = 8 \) whose SNR threshold is 0.4668 dB. This is only 0.02 dB away from that of \( C \).

\[
\lambda(x) = 0.3021x + 0.2831x^2 + 0.4148x^8, \\
\rho(x) = 0.243x^5 + 0.757x^6.
\]

Modifying the check node degree distribution to obtain a better performance can be done systematically using the following algorithm which is the parallel of Algorithm 7-3 in the previous section. In this algorithm for any given check node degree distribution we perform the steps of Algorithm 7-4 while keeping the value of \( D_v \) fixed.

Algorithm 7-5:

Assume that the code rate \( R^* \) and the maximum variable node degree \( D_v \) are given. Also consider the check node degree distribution \( \rho(x) = ax^{D_c-2} + (1-a)x^{D_c-1} \).
1- Compute the highest channel noise power \( \sigma^2_{\text{Shannon}} \) for the reliable transmission at the given rate using the Shannon capacity of the BIAWGN channel. Set \( D_c \) to a large value (for the cases considered here, 9 is appropriate) and \( \alpha \) to 0.

2- Set the channel noise parameter \( \sigma^2 = \sigma^2_{\text{Shannon}} \).

3- Deterministically construct an ensemble by setting

\[
\lambda_i = \lambda_{i,\text{max}} - .001, \quad i = 2,3 \quad \text{and} \quad \lambda_{Dv} = 1 - \sum_{i=2}^{3} \lambda_i.
\]

4- Compute \( R \), the rate of the resulting ensemble and compare it with \( R^* \).

5- If \( |R - R^*| < q \), where \( q \) is the desired precision, go to the next step. Otherwise, if \( R < R^* \), decrease \( \sigma^2 \), and if \( R > R^* \), increase \( \sigma^2 \). The step size in changing \( \sigma^2 \) depends on \( q \). In both cases go to Step 3.

6- Check if the resulting ensemble is convergent, if it is, go to Step 7; otherwise decrease the average check node degree (see Algorithm 7-3 for more details) and go to Step 2.

7- The current \((\lambda(x), \rho(x))\) is the desired ensemble.

In Algorithm 7-5, similar to Algorithm 7-4, for any given check node degree distribution, we look for a channel noise power for which the constructed ensemble has the desired code rate. The resulting ensemble if convergent for the selected channel noise power, is the desired ensemble. Beginning with a large average check node degree, the first constructed ensemble should be non-convergent (if it is convergent, we need to increase the average check node degree and repeat the algorithm). We then start decreasing the average check node degree and repeat a similar procedure. The largest average check node degree for which the constructed ensemble is convergent will have
the best threshold and is the algorithm output. If we continue decreasing the average
check node degree, other convergent ensembles will be constructed. These ensembles
have a higher SNR threshold and do not seem useful for the problem at hand. We will
show however in the next section that these ensembles can be attractive candidates for
transmission of the information over the BIAWGN channel in the presence of channel
estimation error.

The complexity of Algorithm 7-5 is almost the same as that of Algorithm 7-3. Note that in Algorithm 7-5, for each tested check node degree distribution, we perform
density evolution once to see if the designed ensemble is convergent. Once we obtain a
convergent ensemble, the algorithm stops and the channel noise variance $\sigma^2$ based on
which the code is constructed, is the threshold of the ensemble. The SNR threshold of the
ensemble in dB is equal to $-10 \log_{10}(2R^2\sigma^2)$. In an optimization-based design method,
for any candidate pair of degree distributions, the threshold of the code has to be
calculated. This requires performing density evolution several times which makes the
procedure far more complex than ours.

We use Algorithm 7-5 to design ensembles with rate one-half. Our experience
shows that good ensembles with maximum variable node degree less than 10 can be
designed using this algorithm. For larger values of $D_v$ we usually require one extra
constituent variable node degree $j$ (e.g., 6, 7) to obtain a close-to-optimum ensemble. For
$D_v > 10$ we fix $\lambda_j$ at different values and for each value, we perform Algorithm 7-5.
Among the generated ensembles, we select the one with the smallest SNR threshold$^{20}$.

---

$^{20}$ A systematic process has been proposed in the next section for a more general case where SNR mismatch is present.
In Table 7-1, we have reported the results of Algorithm 7-5 for rate one half ensembles with maximum variable node degrees ranging from 7 to 12 and 20. For each ensemble, we have shown the difference between the SNR threshold value of the designed ensemble and that of a similar ensemble in Table I of [7]. Both ensembles have the same value of $D_v$. As can be seen, except for $D_v = 7^{21}$, the SNR-thresholds for our ensembles are at most .05dB away from those of [7]. Note that for $D_v > 9$ we have used extra variable node degrees 6 and 7 for $D_v = 10$, 11, and $D_v = 12, 20$, respectively. Based on our experience similar results can be obtained if we use other degrees which are roughly in the middle of the range between 3 and $D_v$ (e.g., 5, 6, 7, 8).

7.5 Deterministic design of LDPC codes for a given code rate in the presence of mismatch

In this section, we use our findings in previous sections to design codes in the presence of mismatch. Assume that for a given maximum variable node degree, we use the techniques of Section 6.3 to deterministically design an ensemble. We call such an ensemble $C_{no-mismatch}$ with degree distribution pair $(\lambda_{no-mismatch}(x), \rho_{no-mismatch}(x))$. The results of Chapter 4 suggest that while ensemble $C_{no-mismatch}$ is a good candidate in the absence of mismatch, it might be too sensitive to mismatch. Now consider a case where we are interested in designing an ensemble with the given set of constituent variable node degrees 2, 3, $D_v$, and if necessary on extra degree $j$ smaller than $D_v$, that has the lowest

---

21 By subtracting 0.01 from $\lambda_{2,\text{max}}$ and $\lambda_{3,\text{max}}$ in Step 3 in Algorithm 6-5, we were able to design a new ensemble with variable node degree distribution $\lambda(x) = .3115x + .2954x^2 + .3931x^5$ and check node degree distribution $\rho(x) = .518x^5 + .482x^6$. The threshold for this ensemble is 0.5432 dB which is 0.0279 dB away from that of [7].
maximum threshold over the SNR offset interval of \([-\eta, \eta]\) (\(\eta > 0\)). From Section 4.2 we know that this design problem is equivalent to minimizing the ensemble threshold at the SNR offset \(\eta\) (\(\alpha(\eta)\) from Chapters 3 and 4). To tackle this problem using a deterministic approach, we propose the following algorithm. This algorithm is in fact an extended version of Algorithm 7-5.

Table 7-1: Deterministically optimized degree distributions for rate one half ensembles with different maximum variable node degrees.

<table>
<thead>
<tr>
<th>(D_v)</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda_2)</td>
<td>0.3520</td>
<td>0.3021</td>
<td>0.2785</td>
<td>0.2687</td>
<td>0.2565</td>
<td>0.2528</td>
<td>0.2293</td>
</tr>
<tr>
<td>(\lambda_3)</td>
<td>0.3258</td>
<td>0.2831</td>
<td>0.2638</td>
<td>0.2562</td>
<td>0.2461</td>
<td>0.2435</td>
<td>0.2196</td>
</tr>
<tr>
<td>(\lambda_4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_6)</td>
<td></td>
<td></td>
<td>0.0450</td>
<td>0.0500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_7)</td>
<td>0.3223</td>
<td></td>
<td></td>
<td>0.1050</td>
<td>0.2350</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_8)</td>
<td>0.4148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_9)</td>
<td></td>
<td>0.4576</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_{10})</td>
<td></td>
<td></td>
<td>0.4301</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_{11})</td>
<td></td>
<td></td>
<td></td>
<td>0.4473</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_{12})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3987</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lambda_{20})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.4601</td>
<td></td>
</tr>
<tr>
<td>(\rho_5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\rho_6)</td>
<td>0.943</td>
<td>0.243</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\rho_7)</td>
<td>0.057</td>
<td>0.757</td>
<td>0.786</td>
<td>0.568</td>
<td>0.261</td>
<td>0.162</td>
<td></td>
</tr>
<tr>
<td>(\rho_8)</td>
<td></td>
<td>0.214</td>
<td>0.432</td>
<td>0.739</td>
<td>0.838</td>
<td>0.540</td>
<td></td>
</tr>
<tr>
<td>(\rho_9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.460</td>
<td></td>
</tr>
<tr>
<td>Threshold(dB)</td>
<td>0.6608</td>
<td>0.4668</td>
<td>0.4312</td>
<td>0.4271</td>
<td>0.4244</td>
<td>0.4227</td>
<td>0.3438</td>
</tr>
<tr>
<td>Difference with <a href="dB">7</a></td>
<td>0.1455</td>
<td>0.0185</td>
<td>0.0222</td>
<td>0.0344</td>
<td>0.0445</td>
<td>0.0500</td>
<td>0.0334</td>
</tr>
</tbody>
</table>

**Algorithm 7-6:**

Assume that the code rate \(R\), an initial check node degree distribution \(\rho(x)\), and the maximum variable node degree \(D_v\) are given.
1- Compute the largest possible channel noise power $\sigma^2_{\text{Shannon}}$ for the reliable transmission with given rate $R^*$ using the Shannon capacity of BIAWGN channel.

2- Set $\lambda_j = 0$.

3- Set the channel noise parameter $\sigma^2 = \sigma^2_{\text{Shannon}}$.

4- Construct an ensemble by setting

$$\lambda_i = \lambda_{i,\text{max}} - 0.001, \quad i = 2, 3 \quad \text{and} \quad \lambda_{Dv} = 1 - \sum_{i=2}^{3} \lambda_i - \lambda_j$$

5- Compute $R$, the rate of the resulting ensemble and compare it with $R^*$.

6- If $|R - R^*| < q$, where $q$ is the desired precision, go to the next step. Otherwise, if $R < R^*$, decrease $\sigma^2$, and if $R > R^*$, increase $\sigma^2$. The step size in changing $\sigma^2$ depends on $q$. In both cases go to Step 4.

7- If $\lambda_j = 0$ and $\lambda_{Dv} < 0$, go to Step 11 (algorithm ends).

8- If $\lambda_j \neq 0$ and $\lambda_{Dv} < 0$, go to Step 10.

9- Compute the value of $a(\eta_j)$ and store it. Also increase $\lambda_j$ by the step size. Go to Step 3.

10- Decrease the average check node degree by modifying $\rho(x)$ (see Algorithm 7-3 for more details). Go to Step 3.

11- The degree distribution pair with the lowest $a(\eta)$ is the algorithm output.

We have to choose $\rho(x)$ with a large $\bar{d}_e$ (as explained in Algorithm 7-3) as the initial check node degree distribution. For any given check node degree distribution, Steps 3 to 9 of the algorithm are performed and the values of $a(\eta_j)$ are stored. When in
Step 8: $\lambda_{\rho_v} < 0$, Step 10 is performed in which we decrease $\bar{d}_c$. The algorithm continues until the value of $\bar{d}_c$ becomes very small. This makes the values of $\lambda_{2,\max}$ and $\lambda_{3,\max}$ large enough such that the two conditions in Step 7 of the algorithm are satisfied. In this case the algorithm stops. Our experience shows that to decrease the running time of the algorithm, it is advantageous to use $\rho_{\text{no-mismatch}}(x)$ as the initial check node degree distribution.

Example 7-5: We consider a problem similar to that of Chapter 4 where we would like to design a rate one half code with maximum variable node degree 20 whose performance is optimal in the SNR offset interval of [-2,2]. As a point of reference we again consider code ensemble $C_I$ from Chapter 4, designed in [36]. The threshold values for $C_I$ in the presence of mismatch for $\eta$ ranging from -6 to 6dB are shown in Figure 7-2 a starred solid line. Using Algorithm 7-6 we design the following ensemble $C_{\text{mismatch}}$:

$$\lambda_{\text{mismatch}}(x) = .243x + .2351x^2 + .205x^6 + .3169x^{19},$$

$$\rho_{\text{mismatch}}(x) = .820x^6 + .180x^7.$$

The thresholds for this ensemble are shown in Figure 7-2 with squared solid line. As can be seen, in the offset range of [-2dB, 2dB], the maximum threshold has decreased by about 0.2 dB compared to $C_I$. This is at the expense of close to 0.2dB degradation of performance at offset zero. This degradation is mainly because we have optimized the code for $\eta = -2$ dB and partly because of the suboptimal nature of our algorithm.
7.6 Concluding remarks

In this chapter we proposed a deterministic approach to design LDPC codes for BIAWGN channel. We considered both categories of code design (code design for a given channel noise variance and code design for a given rate). In both cases we showed that the ensembles constructed based on our approach perform very closely to those designed by optimization. We then applied the same idea for designing codes in the presence of mismatch. In particular we were able to obtain similar results to those of Chapter 4 in which, using EXIT charts, an ensemble that performed well in the presence of mismatch had been designed.

Figure 7-2: BP thresholds for ensembles $C_j$ and $C_{\text{mismatch}}$ for different offset values.
Chapter 8 Conclusion and Future Work

In this thesis, we investigated the performance of belief propagation for the decoding of regular and irregular LDPC codes in the presence of channel estimation error. It is shown in Chapter 3 that at the limit of infinite SNR mismatch, the performance of belief propagation approaches to that of min-sum and the channel BER, for the over- and the underestimation, respectively. This provides a proof that asymptotically belief propagation is more sensitive to the underestimation of SNR than it is to the overestimation of it. It is however observed that for some ensembles of LDPC codes and over finite ranges of SNR mismatch, overestimation can be almost as harmful as underestimation. We demonstrated that the sensitivity of belief propagation to the SNR mismatch increases with increasing the variable degree ($d_v$) and decreasing the check
degree ($d_c$) for regular LDPC codes. The effect of variable degree is more prominent such that among the codes with the same rate, those with smaller $d_v$ and $d_c$ are less sensitive to mismatch. For irregular LDPC codes, we showed that highly irregular codes, which perform very well in the absence of mismatch, face significant degradation in their performance in the presence of mismatch such that they may perform worse than codes with less irregularity. For the asymptotic analysis, we proposed a method based on the Gaussian approximation of density evolution which tracks both the mean and the variance of message distributions. The method is much simpler than density evolution and the threshold values computed based on this method were shown to match the finite-length simulations.

Based on the results of Chapter 3, in Chapter 4 we designed irregular LDPC codes that were robust against channel estimation errors, when decoded by belief propagation algorithm. In a min-max sense, the designed codes outperformed irregular codes, conventionally optimized for zero SNR mismatch, over the SNR offset range of interest. The performance improvement over the conventional irregular codes in many cases was maintained even outside the mismatch range of interest. We proposed a method to develop the EXIT curves in the presence of mismatch that were used for the code design.

The design methods proposed in [7][8][29][36] as well as that of Chapter 4 are all based on numerical optimization. In Chapter 5, we proposed a method to deterministically design codes for BEC. It was shown that the designed ensembles perform close to the ensembles designed by exhaustive search (or optimization algorithms with asymptotic analysis tools). This is while the design complexity of the proposed ensembles is considerably less than those, obtained by exhaustive search.
Based on the ensembles proposed in Chapter 5, in Chapter 6 we presented new sequences of capacity achieving LDPC code ensembles. We showed that such sequences can give as good performance as those proposed in [22] while they require considerably less number of constituent variable node degrees and smaller maximum variable node degrees. We also showed that with proper choice of \( f(N) \), such sequences are asymptotically quasi-optimal. We also proved that by choosing \( f(N) \) as a linear function of \( N \), our proposed sequences are asymptotically optimal.

In Chapter 7 we extended the results of Chapter 5 to BIAWGN channel. We initially developed the upper bound on degree 3 variable nodes. We then used it together with upper bound on degree 2 variable nodes to design LDPC codes for the BIAWGN channel. We showed that the ensembles constructed based on our approach perform very closely to those designed by optimization. We then applied the method to the case with channel estimation error. We showed that codes that perform well in the presence of mismatch can be designed based on our deterministic approach.

As a possible future work, one can extend the results of Chapter 3 and Chapter 4 to analyze the performance of Bit-Interleaved Low-Density Parity-Check Coded Modulation with Iterative Decoding in the presence of mismatch. A Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID) system consists of a non-binary modulation demapper and a soft-input soft-output decoder that exchange extrinsic information. It was initially introduced in [40] using a Convolutional code in the decoder part. The use of LDPC code in the decoder was suggested in [40][41][42]. In [42] the performance of BICM-ID systems has been investigated in the presence of mismatch using EXIT charts. To generate EXIT charts in [43], it is assumed that the messages can
be modeled as Gaussian variables that have a fixed ratio of variance to mean throughout iterations. We know that this assumption in general is not necessarily valid. The method proposed in Chapter 4 can be used to generate the true EXIT curves of the demapper and the decoder. By comparing our results to those of [43], one can evaluate the accuracy of the assumption of [43].

Most of the results of Chapter 6 are proven for check regular codes. Extending our results for check regular ensemble sequences to the ensembles with other types of check node degree distributions is another possible future work. In particular Tornado sequences [46][22] can be considered.

In Chapter 7 the following important questions remained unanswered that can be very interesting topics of future research:

1-Are there upper bounds for other variable node degrees similar to degree 2 and 3 variable node degrees?

2-For degree 3 variable nodes we derived upper bounds assuming that the check node has the form of (7-2). Does similar upper bound exist for other forms of check node degree distributions?

3- Can an analytical relationship between $\lambda_{3,\text{max}}$, channel noise variance $\sigma^2$ and general check node degree distribution $\rho(x)$, similar to (7-1) be found and proved?
Appendix A: Proof of the Chapter 5 lemmas and theorems

Proof of Lemma 5-1:

To prove claim 1, we show $1/\bar{d}_v' > 1/\bar{d}_v$:

$$1/\bar{d}_v' = \sum_{i=2}^{D_v} \lambda_i / i + (\lambda_a - k) / a + (\lambda_b + k) / b > \sum_{i=2}^{D_v} \lambda_i / i + \lambda_a / a + \lambda_b / b = 1/\bar{d}_v.$$  

For claim 2, we show that the degree distribution for $C'$ satisfies (2-21) For $0 < x \leq 1$, $a < b$, and $k \geq 0$, it is easy to see that $k x^{b-1} - k x^{a-1} \leq 0$. As a result:

$$\lambda_a x^{a-1} + \lambda_b x^{b-1} \geq (\lambda_a - k)x^{a-1} + (\lambda_b + k)x^{b-1} = \lambda_a x^{a-1} + \lambda_b x^{b-1},$$

and therefore $\lambda(x) \geq \lambda'(x)$. This implies that if $\lambda(x)$ satisfies (2-21), so does $\lambda'(x)$, and thus $C'$ is convergent.

Proof of Theorem 5-1:

From (5-2), we have $\sum_{i=2}^{\infty} T_i = 1$, and since $T_2 \leq \varepsilon < 1$, and $T_i > 0, \forall i$, it is easy to see that there exists an integer $N$ that satisfies both (5-6) and (5-7) and that such $N$ is unique. For such $N$, to show the convergence of the Type-A ensemble, we verify (2-21). Substituting $\lambda_i = T_i / \varepsilon$, $2 \leq i \leq N - 1$ in (2-21), we obtain:

$$(\varepsilon \lambda_N - T_N)x^{N-1} - \sum_{i=N+1}^{\infty} T_i x^{i-1} < 0, \ 0 < x \leq 1.$$  

To show that the above inequality holds, we note that $\varepsilon \lambda_N = \varepsilon - \sum_{i=2}^{N-1} T_i$, and thus $\varepsilon \lambda_N - T_N = \varepsilon - \sum_{i=2}^{N} T_i < 0$, where the last inequality is the same as (5-6). Multiplying
both sides by the positive value $x^{N-1}$, and subsequently subtracting the positive value of 
\[ \sum_{i=N+1}^{\infty} T_i x^{i-1} \]
from the left hand side proves (2-21) and thus the convergence.

To prove that $\lambda_N$ is nonnegative from (5-7), we use 
\[ \sum_{i=2}^{N-1} T_i \leq \varepsilon, \]
and thus 
\[ \sum_{i=2}^{N-1} \lambda_i \leq 1 \]
which implies $\lambda_N \geq 0$. ■

**Proof of Theorem 5-2:**
To prove that the channel parameter $\varepsilon$ is the threshold of the Type-A ensemble $C$, we show that if $C$ converges over a channel with parameter $\varepsilon'$, we must have $\varepsilon \geq \varepsilon'$. Based on (5-4), channel parameter $\varepsilon'$ must satisfy $\lambda_2 \leq T_2 / \varepsilon'$. By the construction of $C$, however, $\lambda_2 = T_2 / \varepsilon$, and therefore $T_2 / \varepsilon \leq T_2 / \varepsilon'$. This requires that $\varepsilon \geq \varepsilon'$ ■

**Proof of Proposition 5-1:**
We denote the variable node degree distribution for ensemble $C$ with $\lambda(x)$. For the given check node degree distribution and channel parameter, we construct the Type-A ensemble $C'$ and denote its variable node degree distribution by $\lambda'(x)$. The maximum variable node degree for ensemble $C'$ is $N$. We define functions $f$ and $g$ as follows:

\[ f(x) = \varepsilon x + \sum_{i=2}^{N-1} T_i x^{i-1} + (\varepsilon - \sum_{i=2}^{N-1} T_i) x^{N-1}, \]

\[ g(x) = \varepsilon x + \sum_{i=2}^{P} T_i x^{i-1} + (\varepsilon - \sum_{i=2}^{P} T_i) x^{N-1}. \]

It is easy to see that $f(0) = g(0) = 0$ and $f(1) = g(1) = \varepsilon$. Moreover, based on (2-21) we have:
\[ f(x) < 1 - \rho^{-1}(1-x), \quad 0 < x \leq 1. \]  \hspace{1cm} (A-1)

Taking the derivatives of \( f \) and \( g \), we have

\[ f'(x) = \sum_{i=2}^{N-1} (i-1)T_i x^{i-2} + (N-1)(\varepsilon - \sum_{i=2}^{N-1} T_i)x^{N-2}, \]

\[ g'(x) = \sum_{i=2}^{P} (i-1)T_i x^{i-2} + (D_v - 1)(\varepsilon - \sum_{i=2}^{P} T_i)x^{P-2}. \]

From (5-8) we have:

\[(N - D_v)(\varepsilon - \sum_{i=2}^{P} T_i) \leq \sum_{i=P+1}^{N-1} (N - i)T_i = -\sum_{i=P+1}^{N-1} (i-1)T_i + \sum_{i=P+1}^{N-1} (N-1)T_i.\]

Slight manipulation of this inequality results in

\[(N-1)\left[\varepsilon - \left(\sum_{i=2}^{N-1} T_i - \sum_{i=P+1}^{N-1} T_i\right)\right] - (D_v - 1)(\varepsilon - \sum_{i=2}^{P} T_i) \leq \sum_{i=2}^{P} (i-1)T_i - \sum_{i=2}^{P} (i-1)T_i + \sum_{i=P+1}^{N-1} (N-1)T_i,\]

and consequently

\[(N-1)(\varepsilon - \sum_{i=2}^{N-1} T_i) + (N-1)(\sum_{i=P+1}^{N-1} T_i) - (D_v - 1)(\varepsilon - \sum_{i=2}^{P} T_i) \leq \sum_{i=2}^{P} (i-1)T_i - \sum_{i=2}^{P} (i-1)T_i + (N-1)\sum_{i=P+1}^{N-1} T_i\]

Canceling the term \((N-1)\sum_{i=P+1}^{N-1} T_i\) from both sides and rearranging the rest of the terms, we have

\[\sum_{i=2}^{N-1} (i-1)T_i + (N-1)(\varepsilon - \sum_{i=2}^{N-1} T_i) \leq \sum_{i=2}^{P} (i-1)T_i + (D_v - 1)(\varepsilon - \sum_{i=2}^{P} T_i),\]

which is equivalent to \(f'(1) < g'(1)\), and therefore \(g(x) < f(x)\) in the vicinity of \(x = 1\) for \(x < 1\). To compare the values of \(g(x)\) and \(f(x)\) in the vicinity of \(x = 0\), we start from lower powers of \(x\) in the Taylor series for both functions at \(x = 0\), and note that both series have the same coefficients for powers of \(x\) up to \(P-1\). For \(x^P\), the coefficient for \(g(x)\) is zero.
while for $f(x)$, it is $T_{P+1}$. This implies that $g(x) < f(x)$ in the vicinity of $x = 0$. As $f$ and $g$ are convex functions we conclude that $g(x) < f(x)$ in the whole $(0, 1)$ interval. This combined with (A-1) proves that ensemble $C$ is convergent. ■

**Proof of Theorem 5-3:**

Since $R > 0$, we have $d_v < d_c$, and thus

$$\int_0^1 \rho(x) dx < d_v^{-1}.$$  

Also note that for $0 < x < 1$, $0 < \rho(x) < 1$ and therefore it can be verified that $\int_0^1 \rho(x) dx = 1 - \int_0^1 \rho^{-1}(x) dx$. We therefore have

$$1 - \int_0^1 \rho^{-1}(1-x) dx = \int_0^1 \rho^{-1}(x) dx = \int_0^1 \rho(x) dx < d_v^{-1}. \quad (A-2)$$

Consider the following sequences for $n \geq 2$:

$$f(n) = d_v^{-1} \sum_{i=2}^{n} T_i$$

$$g(n) = \sum_{i=2}^{n} T_i / i$$

To prove Theorem 5-3, we prove that there exists a unique $N$ that satisfies $f(N) > g(N)$ and $f(N-1) \leq g(N-1)$.

Both sequences $f$ and $g$ are strictly increasing. We note that based on inequality $R < 1 - 2 / d_c$, we have $d_v^{-1} < 1 / 2$. Also, $f(2) = d_v^{-1} T_2$ and $g(2) = T_2 / 2$. We thus have $f(2) < g(2)$. Moreover, we have

---

22 Here, we make the assumption that $D_v > P+1$. Otherwise, $N = D_v$ and the Type-MB ensemble will be identical to Type-A ensemble.
\[ f(n) = \bar{d}_v^{-1} \sum_{i=2}^{n} T_i = \bar{d}_v^{-1} \sum_{i=2}^{n} T_i x^{i-1}\bigg|_{x=1} , \]

\[ g(n) = \sum_{i=2}^{n} T_i / i = \sum_{i=2}^{n} (T_i x^i) / i\bigg|_{x=1} . \]

Based on (5-2), this results in

\[ \lim_{n \to \infty} f(n) = \bar{d}_v^{-1} (1 - \rho^{-1}(1 - x))\bigg|_{x=1} = \bar{d}_v^{-1} , \]

and

\[ \lim_{n \to \infty} g(n) = \int_{0}^{1} 1 - \rho^{-1}(1 - x) dx = 1 - \int_{0}^{1} \rho^{-1}(1 - x) dx < \bar{d}_v^{-1} , \]

where the last inequality is based on (A-2). We thus have

\[ \lim_{n \to \infty} f(n) > \lim_{n \to \infty} g(n) . \]

Putting this together with \( f(2) < g(2) \), we conclude that there exist some \( n > 2 \), for which \( f(n) > g(n) \).

Let \( N \) be the smallest such \( n \). We thus have

\[ f(n) \leq g(n) , \quad 2 \leq n \leq N - 1. \] (A-3)

It is also easy to see that \( N \) satisfies

\[ \bar{d}_v^{-1} > 1 / N , \] (A-4)

because otherwise, we have

\[ \bar{d}_v^{-1} T_i \leq T_i / N \leq T_i / i , \quad \text{for} \quad 2 \leq i \leq N \]

and therefore

\[ \bar{d}_v^{-1} \sum_{i=2}^{N} T_i \leq \sum_{i=2}^{N} T_i / i . \]

This means that \( f(N) \leq g(N) \), which contradicts assumption (5-12).
Now we prove that for any \( n > N, f(n) > g(n) \). Using (A-4), we have

\[
\tilde{d}_v^{-1} T_i > T_i / N > T_i / i, \text{ for } i > N,
\]

(A-5)

and thus for any \( n > N \), we have

\[
\tilde{d}_v^{-1} \sum_{i=N+1}^{n} T_i > \sum_{i=N+1}^{n} T_i / i.
\]

This together with (5-12) results in

\[
\tilde{d}_v^{-1} \sum_{i=1}^{n} T_i > \sum_{i=2}^{n} T_i / i,
\]

(A-7)

which is equivalent to \( f(n) > g(n) \). This proves the uniqueness of \( N \).

**Proof of Theorem 5-4:**

Note that based on the proof Theorem 5-3, the value of \( N \) that satisfies (5-12) and (5-13) is unique and using (A-3), for any \( P < N \), we have

\[
\tilde{d}_v^{-1} \sum_{i=2}^{P} T_i < \sum_{i=2}^{P} T_i / i.
\]

(A-8)

Also,

\[
\lambda_N = 1 - \sum_{i=2}^{P} \lambda_i = \frac{1}{\varepsilon} (\varepsilon - \sum_{i=2}^{P} T_i)
\]

\[
= \frac{1}{\varepsilon} \left( \sum_{i=2}^{P} T_i (1/i - 1/N) - [(1-R)^{-1} \tilde{d}_e^{-1} - 1/N] \sum_{i=2}^{P} T_i \right)
\]

\[
= \frac{1}{\varepsilon (\tilde{d}_v^{-1} - 1/N)} \left( \sum_{i=2}^{P} T_i / i - \tilde{d}_v^{-1} \sum_{i=2}^{P} T_i \right)
\]

Based on (A-8) and that \( \tilde{d}_v^{-1} > 1/N \), we conclude that \( \lambda_N > 0 \).
Proof of Theorem 5-5:

Let $C$ denote the Type-B ensemble. This implies that $P < N-1$. For the given check node degree distribution and code rate, let $C'$ denote the corresponding Type-A ensemble. Note that $C'$ is convergent on a channel with parameter equal to (5-10). Denote the threshold of ensemble $C'$ by $\varepsilon'$. We have

$$\lambda(x) = \sum_{i=2}^{p} (T_i / \varepsilon)x^{i-1} + (1 - \sum_{i=2}^{p} T_i / \varepsilon)x^{N-1},$$

and

$$\lambda'(x) = \sum_{i=2}^{N-1} (T_i / \varepsilon')x^{i-1} + (1 - \sum_{i=2}^{N-1} T_i / \varepsilon')x^{N-1}.$$ 

Therefore

$$\varepsilon \lambda(x) = \sum_{i=2}^{p} T_i x^{i-1} + (\varepsilon - \sum_{i=2}^{p} T_i)x^{N-1},$$

and

$$\varepsilon' \lambda'(x) = \sum_{i=2}^{N-1} T_i x^{i-1} + (\varepsilon' - \sum_{i=2}^{N-1} T_i)x^{N-1}.$$ 

Ensemble $C'$ is convergent by definition, therefore

$$\varepsilon' \lambda'(x) < 1 - \rho^{-1}(1-x), \text{ for } 0 < x \leq 1.$$ 

(A-11)

We also have

$$\varepsilon = \frac{\sum_{i=2}^{p} T_i (1/i - 1/N)}{(1 - R)^{-1} d_c^{-1} - 1/N} = \frac{\sum_{i=2}^{N-1} T_i (1/i - 1/N)}{(1 - R)^{-1} d_c^{-1} - 1/N} = \varepsilon',$$

since $P < N$ and $1/i > 1/N$ for $i \leq N-1$. Therefore $\varepsilon < \varepsilon'$, and consequently

$$\sum_{i=2}^{N-1} T_i x^{i-1} + (\varepsilon - \sum_{i=2}^{N-1} T_i)x^{N-1} < \sum_{i=2}^{N-1} T_i x^{i-1} + (\varepsilon' - \sum_{i=2}^{N-1} T_i)x^{N-1}.$$ 

(A-12)
Also, as $0 < x \leq 1$, one can see that $T_i x^{i-1} - T_i x^{N-1} \geq 0$, for $P + 1 \leq i \leq N - 1$, and thus

$$
\sum_{i=P+1}^{N-1} T_i x^{i-1} - \sum_{i=P+1}^{N-1} T_i x^{N-1} \geq 0.
$$
(A-13)

Adding (A-13) to (A-9), we obtain

$$
\epsilon \lambda(x) \leq \sum_{i=2}^{P} T_i x^{i-1} + (\epsilon - \sum_{i=2}^{P} T_i) x^{N-1} + \sum_{i=P+1}^{N-1} T_i x^{i-1} - \sum_{i=P+1}^{N-1} T_i x^{N-1},
$$

and therefore

$$
\epsilon \lambda(x) \leq \sum_{i=2}^{N-1} T_i x^{i-1} + (\epsilon - \sum_{i=2}^{N-1} T_i) x^{N-1}.
$$
(A-14)

Combining (A-14), (A-12) and (A-10), we have

$$
\epsilon \lambda(x) \leq \sum_{i=2}^{N-1} T_i x^{i-1} + (\epsilon - \sum_{i=2}^{N-1} T_i) x^{N-1} < \sum_{i=2}^{N-1} T_i x^{i-1} + (\epsilon' - \sum_{i=2}^{N-1} T_i) x^{N-1} = \epsilon' \lambda'(x),
$$

and thus

$$
\epsilon \lambda(x) < \epsilon' \lambda'(x).
$$
(A-15)

From (A-11) and (A-15) we conclude that

$$
\epsilon \lambda(x) < 1 - \rho^{-1}(1 - x), \text{ for } 0 < x \leq 1
$$

which proves the convergence. With an argument similar to that of Theorem 5-2, one can show that (5-14) is in fact the threshold of the Type-B ensemble.
Appendix B: Fractional Binominal Coefficients

Based on the properties of the Fractional Binominal Coefficients in [21], we can write the following relationships for $0 < \alpha < 1$:

\[
\sum_{i=1}^{N} \binom{\alpha}{i} (-1)^{i-1} = \sum_{i=2}^{N} \binom{\alpha}{i-1} (-1)^{i-1} = \sum_{i=2}^{N} T_i = 1 - \frac{N}{\alpha} \left[ \frac{\alpha}{N} \right] \tag{B-1}
\]

\[
\sum_{i=1}^{N-1} \binom{\alpha}{i} \frac{(-1)^{i+1}}{i+1} = \sum_{i=2}^{N} \binom{\alpha}{i-1} \frac{(-1)^{i}}{i} = \sum_{i=2}^{N} \frac{T_i}{i} = \frac{\alpha - \left[ \frac{\alpha}{N} \right]}{\alpha + 1} \tag{B-2}
\]

Also from the proof of Proposition I of [21], we have:

\[
\frac{\alpha}{N^{\alpha+1}} e^{\alpha(2 - \gamma - 1/2N)} (1 - \alpha)^2 \leq \left[ \frac{\alpha}{N} \right] \leq \frac{\alpha}{N^{\alpha+1}} e^{\alpha(1 - \gamma)} (1 - \alpha) \tag{B-3}
\]

where $\gamma$ is the Euler constant.
Appendix C: Proof of the Chapter 6 lemmas and theorems

Proof of Theorem 6-1:
Refer to the proof of Theorem 5-1. ■

Proof of Theorem 6-2:
Refer to the proof of Theorem 5-3, 5-4 and 5-5. ■

Proof of Lemma 6-1:
For GC ensembles, based on (6-5) $N$ has to be chosen such that $\sum_{i=2}^{N} T_i > \varepsilon$. Based on (6-2) it is easy to see that $T_2 \geq T_i$, $2 \leq i \leq N$. Therefore $NT_2 > \varepsilon$. Replacing $T_2$ with $a$, we then have

$$N > \frac{\varepsilon}{a}.$$  

If $D_c$ tends to infinity, $a$ tends to zero and $N$ will tend to infinity. To prove that

$$\lim_{a \to 0} \frac{1}{N^\beta \alpha} = 0,$$

we again start from (6-5) and using (B-1) write

$$1 - \frac{N}{\alpha} \left[ \frac{\alpha}{N} \right] > \varepsilon.$$

Using the lower bound in (B-3), we have

$$1 - \frac{N}{\alpha} \frac{\alpha}{N^\alpha} e^{a(2 - \gamma - 1/2N)} (1 - \alpha)^2 > \varepsilon.$$  

Therefore:

$$\frac{1}{N^\alpha} < \frac{1 - \varepsilon}{e^{a(2 - \gamma - 1/2N)} (1 - \alpha)^2}.$$  

Taking the power $\beta/\alpha$ of both sides and then multiplying both sides by $1/\alpha$, we have

$$\frac{1}{N^\beta \alpha} < \frac{1 - \varepsilon}{\alpha \left( e^{a(2 - \gamma - 1/2N)} (1 - \alpha)^2 \right)^{\beta/\alpha}}.$$  

Also,
\[
\lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{1 - \epsilon}{e^{\alpha(2 - \gamma - 1/2N)}(1 - \alpha)^2} \right)^{\beta/\alpha} = \lim_{\alpha \to 0} \frac{1}{\alpha} (1 - \epsilon)^{\beta/\alpha} = 0.
\]

where the second limit is zero because \(0 < 1 - \epsilon < 1\). Based on (C-1), we therefore have

\[
\lim_{\alpha \to 0} \frac{1}{N^{\beta/\alpha}} \leq \lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{1 - \epsilon}{e^{\alpha(2 - \gamma - 1/2N)}(1 - \alpha)^2} \right)^{\beta/\alpha} = 0.
\]

This completes the proof for GC sequences.

For GR ensembles, based on (6-8) \(N\) has to be chosen such that \(\bar{d}_v^{-1} > \sum_{i=2}^{N} T_i / i \sum_{i=2}^{N} T_i\).

Replacing \(i\) with \(N\) in the denominator of the term \(T_i / i\), we obtain \(\bar{d}_v^{-1} > \frac{\sum_{i=2}^{N} T_i / N}{\sum_{i=2}^{N} T_i} = 1 / N\).

By replacing \(\bar{d}_v^{-1}\) with \(\frac{1}{(1 - R)(1 + 1/\alpha)}\) in this inequality, we have

\[
N > (1 - R)(1 + 1/\alpha).
\]

If \(D_c\) tends to infinity, \(\alpha\) tends to zero and \(N\) will tend to infinity. To prove that

\[\lim_{\alpha \to 0} \frac{1}{N^{\beta/\alpha}} = 0,\]

we again start from (6-8) and using (B-1) and (B-2) write

\[
\frac{\alpha / (\alpha + 1)}{1 - R} > \frac{\alpha + 1}{1 - N \frac{\alpha}{\alpha + 1}}.
\]

Using the upper bound of (B-3) in numerator and the lower bound of (B-3) in denominator, we have

\[
\frac{\alpha / (\alpha + 1)}{1 - R} > \frac{\alpha - e^{\alpha(1 - \gamma)}(1 - \alpha) \frac{\alpha}{N^{\alpha + 1}}}{1 - N \frac{\alpha}{\alpha + 1} e^{\alpha(2 - \gamma - 1/2N)}(1 - \alpha)^2 \frac{\alpha}{N^{\alpha + 1}}} = \frac{\alpha}{\alpha + 1} \frac{1 - e^{\alpha(1 - \gamma)}(1 - \alpha) \frac{1}{N^{\alpha + 1}}}{1 - e^{\alpha(2 - \gamma - 1/2N)}(1 - \alpha)^2 \frac{1}{N^{\alpha}}}
\]
Therefore,

\[
\frac{1}{1 - R} > \frac{N^{\alpha+1} - e^{a(1-\gamma)}(1-\alpha)}{N^{\alpha+1} - Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2} = \frac{N^{\alpha} - e^{a(1-\gamma)}(1-\alpha)/N}{N^{\alpha} - e^{a(2-\gamma-1/2N)}(1-\alpha)^2},
\]

which results in

\[
R > 1 - \frac{N^{\alpha} - e^{a(2-\gamma-1/2N)}(1-\alpha)^2}{N^{\alpha} - e^{a(1-\gamma)}(1-\alpha)/N} = \frac{-e^{a(1-\gamma)}(1-\alpha)/N + e^{a(2-\gamma-1/2N)}(1-\alpha)^2}{N^{\alpha} - e^{a(1-\gamma)}(1-\alpha)/N},
\]

and thus,

\[
N^{\alpha}R > (e^{a(1-\gamma)}(1-\alpha)R - e^{a(1-\gamma)}(1-\alpha)) / N + e^{a(2-\gamma-1/2N)}(1-\alpha)^2.
\]

We then have

\[
\frac{1}{N^{\alpha}} < \frac{R}{(e^{a(1-\gamma)}(1-\alpha)R - e^{a(1-\gamma)}(1-\alpha)) / N + e^{a(2-\gamma-1/2N)}(1-\alpha)^2}.
\]

Taking both sides to the power of \(\beta/\alpha\) and dividing them by \(\alpha\) we obtain

\[
\frac{1}{\alpha N^{\beta}} < \frac{1}{\alpha} \left( \frac{R}{(e^{a(1-\gamma)}(1-\alpha)R - e^{a(1-\gamma)}(1-\alpha)) / N + e^{a(2-\gamma-1/2N)}(1-\alpha)^2} \right)^{\beta/\alpha}.
\]

(C-2)

Taking the limit of the right hand side of (C-2), we have

\[
\lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{R}{(e^{a(1-\gamma)}(1-\alpha)R - e^{a(1-\gamma)}(1-\alpha)) / N + e^{a(2-\gamma-1/2N)}(1-\alpha)^2} \right)^{\beta/\alpha} = \lim_{\alpha \to 0} \frac{1}{\alpha} h(N, \alpha)^{\beta/\alpha}
\]

where

\[
h(N, \alpha) = \frac{NR}{(e^{a(1-\gamma)}(1-\alpha)R - e^{a(1-\gamma)}(1-\alpha)) + Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2} \approx \frac{NR}{(R-1) + N} < 1.
\]

Note that the value of \(h(N, \alpha)\) for large enough \(N\) is always less than 1 and tends to \(R\).

Also note that \(\lim_{\alpha \to 0} (R)^{\beta/\alpha} = 0, \quad 0 < R < 1\). We therefore conclude that
\[
\lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{R}{(e^{\alpha(1-\gamma)}(1-\alpha)R - e^{\alpha(1-\gamma)}(1-\alpha))^N + e^{\alpha(2-\gamma-1/2N)}}(1-\alpha)^2 \right)^{\beta/\alpha} = 0.
\]

Based on (C-2), we then have

\[
\lim_{\alpha \to 0} \frac{1}{N} \frac{R}{\alpha} \leq \lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{R}{(e^{\alpha(1-\gamma)}(1-\alpha)R - e^{\alpha(1-\gamma)}(1-\alpha))^N + e^{\alpha(2-\gamma-1/2N)}}(1-\alpha)^2 \right)^{\beta/\alpha} = 0
\]

This completes the proof for GR sequences.

**Proof of Theorem 6-3:**

For both GC and GR ensembles, we have

\[
\bar{d}_{v_i}^{-1} = \sum_{i=2}^{f(N)} \lambda_i / i + \frac{1}{N} \lambda_N.
\]

Using (2-11), it is then easy to see that the following relationship holds between \(\epsilon\) (the threshold) and \(R\):

\[
\frac{1}{1-R} = \frac{\alpha + 1}{\alpha} \left( \frac{1}{e} \sum_{i=2}^{f(N)} T_i / i + \frac{\lambda_N}{N} \right),
\]

where \(\alpha\) is defined in (6-2). We thus have

\[
1 - \frac{\epsilon}{1-R} = 1 - \frac{\alpha + 1}{\alpha} \left( \sum_{i=2}^{f(N)} T_i / i + \frac{\epsilon \lambda_N}{N} \right) = 1 - \frac{\alpha + 1}{\alpha} \left( \frac{\alpha}{\alpha + 1} \frac{\epsilon}{f(N)} + \frac{\lambda_N}{N} \right),
\]

where for the second equality we have used (B-2). By applying the lower and the upper bounds of (B-3) to \(\frac{\alpha}{f(N)}\) in the above equation, we obtain

\[
\frac{e^{\alpha(2-\gamma-1/2N)}}{f(N)^{\alpha+1}} (1-\alpha)^2 - \frac{\epsilon(\alpha+1)}{\alpha N} \lambda_N \leq 1 - \frac{\epsilon}{1-R} \leq \frac{e^{\alpha(1-\gamma)}(1-\alpha)}{f(N)^{\alpha+1}} - \frac{\epsilon(\alpha+1)}{\alpha N} \lambda_N. \quad \text{(C-3)}
\]
where \( \gamma = .577215... \) is the Euler constant. Taking the limit of both the lower and the upper bounds in (C-3) as \( \alpha \to 0 \) (and \( N \to \infty \)), we observe that they both have the same value:

\[
\lim_{\alpha \to 0} \left[ e^{\alpha(2-\gamma-1/2N)} (1-\alpha)^2 \frac{\alpha + 1}{f(N)^{\alpha+1}} - e^{\lambda_N} \frac{\alpha + 1}{\alpha N} \right] = \lim_{\alpha \to 0} \left[ e^{\alpha(1-\gamma)} (1-\alpha) \frac{\alpha + 1}{f(N)^{\alpha+1}} - e^{\lambda_N} \frac{\alpha + 1}{\alpha N} \right] = \lim_{\alpha \to 0} \frac{1}{f(N)^{\alpha+1}} \lim_{\alpha \to 0} \frac{e^{\lambda_N}}{\alpha N}.
\]

The first limit at the right side of the second equality is zero since \( f(N) \) is a strictly increasing function of \( N \). The second limit is also zero because both \( \varepsilon \) and \( \lambda_N \) are bounded between 0 and 1 and based on Lemma 6-1, \( \lim_{\alpha \to 0} \frac{1}{\alpha N} = 0 \). We therefore conclude that

\[
\lim_{\alpha \to 0} \frac{1}{1-R} = 0
\]

In other words, as \( D_e \) tends to infinity, for a GC ensemble, \( R \) has to tend to \( 1-\varepsilon \), and for a GR ensemble, \( \varepsilon \) has to tend to \( 1-R \). This proves the theorem. ■

**Proof of Lemma 6-2:**

From (C-3) for GC and GR ensemble sequences, we have

\[
\frac{e^{\alpha(2-\gamma-1/2N)}}{f(N)^{\alpha+1}} (1-\alpha)^2 \frac{\alpha + 1}{\alpha N} \lambda_N \leq 1 - \frac{\varepsilon}{1-R} \leq \frac{e^{\alpha(1-\gamma)} (1-\alpha)}{f(N)^{\alpha+1}} \frac{\alpha + 1}{\alpha N} \lambda_N,
\]

but

\[
\lambda_N = 1 - \frac{1}{\varepsilon} \sum_{i=2}^{f(N)} T_i = 1 - \frac{1}{\varepsilon} \left[ 1 - \frac{f(N)}{\alpha} \left[ \frac{\alpha}{f(N)} \right] \right]
\]

Inserting \( \lambda_N \) into (C-3) and replacing \( \left[ \frac{\alpha}{f(N)} \right] \) with its upper bound and lower bound we have:
\[
\frac{(1 - \varepsilon)(\alpha + 1) + e^{a(2-\gamma-1/2N)}(1-\alpha)^2 - \alpha + 1 e^{a(1-\gamma)}(1-\alpha)}{\alpha N f(N)^{\alpha+1}} \leq \frac{1 - \varepsilon}{1 - R} \tag{C-4}
\]

We now take the logarithm of the sides and multiply them by \( D_c^{-1} = \frac{\alpha}{\alpha + 1} \):

\[
\frac{\alpha}{\alpha + 1} \log \left( \frac{(1 - \varepsilon)(\alpha + 1) + e^{a(2-\gamma-1/2N)}(1-\alpha)^2} {\alpha N f(N)^{\alpha+1}} - \frac{\alpha + 1 e^{a(1-\gamma)}(1-\alpha)}{\alpha N f(N)^{\alpha}} \right) \leq D_c^{-1} \log \left( \frac{1 - \varepsilon}{1 - R} \right)
\]

Taking the limit when \( \alpha \to 0 \) and \( N \to \infty \), we observe that the upper bound and lower bound have the same limit:

\[
\lim_{\alpha \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{(1 - \varepsilon)(\alpha + 1) + e^{a(2-\gamma-1/2N)}(1-\alpha)^2} {\alpha N f(N)^{\alpha+1}} - \frac{\alpha + 1 e^{a(1-\gamma)}(1-\alpha)}{\alpha N f(N)^{\alpha}} \right) =
\]

\[
\lim_{\alpha \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{(1 - \varepsilon)(\alpha + 1) + e^{a(1-\gamma)}(1-\alpha)}{\alpha N f(N)^{\alpha+1}} - \frac{\alpha + 1 e^{a(2-\gamma-1/2f(N))}(1-\alpha)^2}{\alpha N f(N)^{\alpha}} \right) =
\]

\[
\lim_{\alpha \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{1}{f(N)^{\alpha+1}} - \frac{1}{\alpha N f(N)^{\alpha}} - (1 - \varepsilon) \right)
\]

We therefore conclude that:

\[
\lim_{\alpha \to 0} \sup D_c^{-1} \log \left( \frac{1 - \varepsilon}{1 - R} \right) = \lim_{\alpha \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{(1 - \varepsilon)(\alpha + 1) + e^{a(1-\gamma)}(1-\alpha)}{\alpha N f(N)^{\alpha+1}} - \frac{\alpha + 1 e^{a(2-\gamma-1/2f(N))}(1-\alpha)^2}{\alpha N f(N)^{\alpha}} \right)
\]

This completes the proof.\( \blacksquare \)

**Proof of Lemma 6-3:**

From (C-4) we had:
\[
\frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(2-\gamma-1/2N)}(1-\alpha)^2}{f(N)^{\alpha+1}} \leq \frac{\alpha+1}{\alpha N} e^{(1-\gamma)}(1-\alpha) \leq 1 - \frac{\varepsilon}{1-R}
\]

\[
\frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(2-\gamma-1/2N)}(1-\alpha)^2}{f(N)^{\alpha+1}} \leq \frac{\alpha+1}{\alpha N} e^{(1-\gamma)}(1-\alpha)
\]

We now multiply the sides by $R^{-(1+1/\alpha)}$

\[
R^{-(1+1/\alpha)} \left( \frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(2-\gamma-1/2N)}(1-\alpha)^2}{f(N)^{\alpha+1}} \right) \leq R^{-(1-\varepsilon)} \left( \frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(1-\gamma)}(1-\alpha)}{f(N)^{\alpha+1}} \right)
\]

Taking the limit when $\alpha \to 0$ and $N \to \infty$, we observe that the upper bound and lower bound have the same limit:

\[
\lim_{\alpha \to 0} R^{-(1+1/\alpha)} \left( \frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(2-\gamma-1/2N)}(1-\alpha)^2}{f(N)^{\alpha+1}} \right) = 1
\]

\[
\lim_{\alpha \to 0} \frac{(1-\varepsilon)(\alpha+1)}{\alpha N} + \frac{e^{(1-\gamma)}(1-\alpha)}{f(N)^{\alpha+1}} \leq 1 - \frac{\varepsilon}{1-R}
\]

\[
\lim_{\alpha \to 0} R^{-(1+1/\alpha)} \left( \frac{1}{f(N)^{\alpha+1}} - \frac{1}{\alpha N} \left( \frac{1}{f(N)^{\alpha}} - (1-\varepsilon) \right) \right) = \lim_{\alpha \to 0} \left( \frac{1}{f(N)^{\alpha+1}} - \frac{1}{\alpha N} \left( \frac{1}{f(N)^{\alpha}} - R \right) \right)
\]

Where the last equality was written because the sequence is capacity achieving and

\[
\lim_{\alpha \to 0} (1-\varepsilon) = R. \text{ We therefore conclude that}
\]

\[
\lim_{\alpha \to 0} \sup \ R^{-(1+1/\alpha)} \left( \frac{1}{f(N)^{\alpha+1}} - \frac{1}{\alpha N} \left( \frac{1}{f(N)^{\alpha}} - R \right) \right)
\]

This completes the proof for GR sequences. The proof for GC sequences is similar.\[\blacksquare\]

**Proof of Lemma 6-4:**

For GC sequences from (C-1) which was obtained based on (6-5) we have:

139

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[
\frac{e^{a(2-\gamma-1/2N)}(1-\alpha)^2}{N^\alpha} < 1 - \varepsilon
\]  
(C-5)

with a similar argument and using (6-6) we can see that:

\[
1 - \varepsilon < \frac{e^{a(1-\gamma)}(1-\alpha)}{(N-1)^\alpha}
\]  
(C-6)

Also note that

\[
\lim_{\alpha \to 0} \frac{1}{N^\alpha} = \lim_{\alpha \to 0} \frac{1}{(N-1)^\alpha}, \text{ therefore:}
\]

\[
\lim_{\alpha \to 0} \frac{1}{N^\alpha} = \lim_{\alpha \to 0} \frac{e^{a(2-\gamma-1/2N)}(1-\alpha)^2}{N^\alpha} = \lim_{\alpha \to 0} \frac{e^{a(1-\gamma)}(1-\alpha)}{(N-1)^\alpha} = \lim_{\alpha \to 0} \frac{e^{a(1-\gamma)}(1-\alpha)}{(N-1)^\alpha}
\]

(C-7)

i.e., the lower bound and upper bound on \(1 - \varepsilon\) have the same limit equal to \(\lim_{\alpha \to 0} \frac{1}{N^\alpha}\).

For (C-5), (C-6) and (C-7) we conclude that:

\[
\lim_{\alpha \to 0} \frac{1}{N^\alpha} = 1 - \varepsilon
\]

If we take the power \(-1/\alpha\) of the sides of the (C-5) and (C-6) and then divide the sides by \(N\) we get:

\[
\left(e^{a(2-\gamma-1/2N)}(1-\alpha)^2\right)^{-1/\alpha} > \frac{(1-\varepsilon)^{-1/\alpha}}{N} > \left(e^{a(1-\gamma)}(1-\alpha)\right)^{-1/\alpha} \frac{N-1}{N}
\]

or

\[
e^{-2+\gamma+1/2N} (1-\alpha)^{-2/\alpha} > \frac{(1-\varepsilon)^{-1/\alpha}}{N} > e^{-1+\gamma} (1-\alpha)^{-1/\alpha} \frac{N-1}{N}
\]

For the upper bound we have:

\[
\lim_{\alpha \to 0} e^{-2+\gamma+1/2N} (1-\alpha)^{-2/\alpha} = e^{-2+\gamma} \lim_{\alpha \to 0} (1-\alpha)^{-2/\alpha} = e^\gamma
\]

(C-8)

For the lower bound we have:

\[
\lim_{\alpha \to 0} e^{-1+\gamma} (1-\alpha)^{-1/\alpha} \frac{N-1}{N} = e^{-1+\gamma} \lim_{\alpha \to 0} (1-\alpha)^{-1/\alpha} = e^\gamma
\]

(C-9)

where for both of the equations we used the following relationship:
\[
\lim_{a \to 0} (1 - \alpha)^{-c/\alpha} = e^c \text{ where } c \text{ is a positive constant.}
\]

From (C-8) and (C-9) we conclude that:

\[
\lim_{a \to 0} \frac{(1 - \epsilon)^{-1/\alpha}}{N} = e^\gamma
\]

This completes the proof for GC sequences. For GR from (6-8) we have:

\[
\frac{\alpha}{\alpha + 1} > \frac{\alpha + 1}{1 - \sigma}
\]

Using the upper bound of (B-3) in numerator and lower bound on (B-3) in denominator, we have:

\[
\frac{\alpha - e^{\alpha(1-\gamma)}}{\alpha(1 - \alpha)} > \frac{\alpha + 1}{1 - \sigma}
\]

Simplifying the equation and inverting the sides we have:

\[
1 - R < \frac{N^{\alpha + 1} - Ne^{\alpha(2-\gamma/2N)}(1 - \alpha)^2}{N^{\alpha + 1} - e^{\alpha(1-\gamma)}(1 - \alpha)}
\]

and therefore:

\[
\frac{Ne^{\alpha(2-\gamma/2N)}(1 - \alpha)^2 - e^{\alpha(1-\gamma)}(1 - \alpha)}{N^{\alpha + 1} - e^{\alpha(1-\gamma)}(1 - \alpha)} < R \quad \text{(C-10)}
\]

With a similar argument and using (6-9) we can see that:

\[
R < \frac{(N - 1)e^{\alpha(1-\gamma)}(1 - \alpha) - e^{\alpha(2-\gamma/2(N-1))}(1 - \alpha)^2}{(N - 1)^{\alpha + 1} - e^{\alpha(2-\gamma/2(N-1))}(1 - \alpha)^2} \quad \text{(C-11)}
\]

Using \( \lim_{a \to 0} \frac{1}{N^\alpha} = \lim_{a \to 0} \frac{1}{(N - 1)^\alpha} \), we write:
\[
\lim_{a \to 0} \frac{1}{N^a} = \lim_{a \to 0} \frac{Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2 - e^{a(1-\gamma)}(1-\alpha)}{N^{a+1} - e^{a(1-\gamma)}(1-\alpha)} \\
= \lim_{a \to 0} \frac{(N-1)e^{a(1-\gamma)}(1-\alpha) - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2}{(N-1)^{a+1} - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2}
\]

i.e. the lower bound and upper bound on \( R \) have the same limit equal to \( \lim_{a \to 0} \frac{1}{N^a} \).

For (C-10), (C-11) and (C-12) we conclude that:

\[
\lim_{a \to 0} \frac{1}{N^a} = R
\]

If we take the power \(-1/\alpha\) of the sides of the (C-10) and (C-11) and then divide the sides by \( N \) we obtain:

\[
\frac{1}{N} \left( \frac{Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2 - e^{a(1-\gamma)}(1-\alpha)}{N^{a+1} - e^{a(1-\gamma)}(1-\alpha)} \right)^{-1/\alpha} > \frac{R^{-1/\alpha}}{N} \\
> \frac{1}{N} \left( \frac{(N-1)e^{a(1-\gamma)}(1-\alpha) - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2}{(N-1)^{a+1} - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2} \right)^{-1/\alpha}
\]

or

\[
\left( \frac{1}{N^{-a}} \frac{Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2 - e^{a(1-\gamma)}(1-\alpha)}{N^{a+1} - e^{a(1-\gamma)}(1-\alpha)} \right)^{-1/\alpha} > \frac{R^{-1/\alpha}}{N} \\
> \left( \frac{1}{N^{-a}} \frac{(N-1)e^{a(1-\gamma)}(1-\alpha) - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2}{(N-1)^{a+1} - e^{a(2-\gamma-1/2(N-1))}(1-\alpha)^2} \right)^{-1/\alpha}
\]

For the upper bound we have:

\[
\lim_{a \to 0} \left( \frac{1}{N^{-a}} \frac{Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2 - e^{a(1-\gamma)}(1-\alpha)}{N^{a+1} - e^{a(1-\gamma)}(1-\alpha)} \right)^{-1/\alpha} \\
= \lim_{a \to 0} \left( \frac{1}{N^{-a}} \frac{Ne^{a(2-\gamma-1/2N)}(1-\alpha)^2}{N^{a+1}} \right)^{-1/\alpha} \\
= \lim_{a \to 0} \left( e^{a(2-\gamma)}(1-\alpha)^2 \right)^{-1/\alpha} = e^{\gamma} \lim_{a \to 0} (e^{-2(a - 2/\alpha)}) = e^\gamma
\]

For the lower bound we have:
\[
\lim_{\alpha \to 0} \left( \frac{1}{N^{1-\alpha}} \left( \frac{N-1}{e^{a(1-\gamma)}} (1-\alpha) - e^{a(2-\gamma-1/2(N-1)) (1-\alpha)^2} \right) \right)^{-1/\alpha} = \\
\lim_{\alpha \to 0} \left( \frac{1}{N^{1-\alpha}} \left( \frac{N-1}{e^{a(1-\gamma)}} (1-\alpha) \right) \right)^{-1/\alpha} = \lim_{\alpha \to 0} \left( \frac{1}{N^{1-\alpha}} e^{a(1-\gamma)} (1-\alpha) \right)^{-1/\alpha} \\
= \lim_{\alpha \to 0} (e^{a(1-\gamma)} (1-\alpha))^{-1/\alpha} = e^{\gamma} \lim_{\alpha \to 0} e^{1/\alpha} (1-\alpha)^{-1/\alpha} = e^{\gamma}
\]

From (C-13) and (C-14) we conclude that:

\[
\lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{N} = e^{\gamma}
\]

This completes the proof for GR sequences.

**Proof of Lemma 6-5:**

First we consider GC ensemble sequences. From Lemma 6-3, \( \lim_{N \to \infty} \frac{1}{N^{1-\alpha}} = 1 - \varepsilon \). We therefore have:

\[
\lim_{\alpha \to 0} \frac{f(N)^\alpha}{N^\alpha} = \lim_{\alpha \to 0} \left( \frac{f(N)}{N} \right)^\alpha = \frac{M}{1 - \varepsilon}, \text{ and thus}
\]

\[
\lim_{\alpha \to 0} \frac{f(N)}{N} = \lim_{\alpha \to 0} \left( \frac{M}{1 - \varepsilon} \right)^{1/\alpha}
\]

Note that based on the lemma assumption, \( \lim_{N \to \infty} \frac{f(N)}{N} < 1 \) and therefore \( \frac{M}{1 - \varepsilon} < 1 \). We have:

\[
\lim_{\alpha \to 0} \frac{f(N)}{N^\alpha} = \lim_{\alpha \to 0} \left( \frac{M}{1 - \varepsilon} \right)^{1/\alpha} = 0
\]
where the last equality is true because $\frac{M}{1-\varepsilon} < 1$. A similar argument can be used for $GR$ ensemble sequences with $\lim_{a \to 0} \frac{1}{N^a} = R$. This completes the proof.

**Proof of Theorem 6-4**: 

We first consider the $GR$ ensemble sequence with code rate $R$. From Lemma 6-2, we have

$$\lim_{a \to 0} \sum_{D_c^{-1} \log \left( 1 - \frac{\varepsilon}{1-R} \right)} = \lim_{a \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{1}{f(N)^{a+1}} - \frac{1}{\alpha N} \left( \frac{1}{f(N)^a} - (1-\varepsilon) \right) \right) \quad (C-15)$$

We now manipulate the limit at the right side of the equation (C-15).

$$\lim_{a \to 0} \frac{\alpha}{\alpha + 1} \log \left( \frac{1}{f(N)^{a+1}} - \frac{1}{\alpha N} \left( \frac{1}{f(N)^a} - (1-\varepsilon) \right) \right) =$$

$$\lim_{a \to 0} \alpha \log \left( \frac{N \alpha - f(N) + (1-\varepsilon) f(N)^{a+1}}{\alpha N f(N)^{a+1}} \right) =$$

$$\lim_{a \to 0} \alpha \log \left( N \alpha - f(N) + (1-\varepsilon) f(N)^{a+1} \right) - \lim_{a \to 0} \alpha \log (\alpha N f(N)^{a+1}) \quad (C-16)$$

For the second limit in the last equation of (C-16), we have

$$\lim_{a \to 0} \alpha \log (\alpha N f(N)^{a+1}) = \lim_{a \to 0} \alpha \log \alpha + \lim_{a \to 0} \alpha \log N + \lim_{a \to 0} \alpha (\alpha + 1) \log f(N)$$

$$= - \log R + \log M. \quad (C-17)$$

where the last equality is true because $\lim_{a \to 0} \alpha \log \alpha = 0$, $\lim_{a \to 0} \alpha \log N = -\log R$ based on Lemma 6-3, and $\lim_{a \to 0} \alpha \log N = \log M$ based on the theorem assumption.

For the first limit in the last equation of (C-16), we initially consider the terms inside the log(•).
\[
\lim_{a \to 0} \left( N\alpha - f(N) + (1-\varepsilon) f(N)^{a+1} \right) = \lim_{a \to 0} N\alpha \left( 1 - \frac{f(N)}{N\alpha} \right) (1 - (1-\varepsilon) f(N)^{a}) = \lim_{a \to 0} N\alpha (1 - (1-\varepsilon) f(N)^{a})
\]

\[
\lim_{a \to 0} \left( 1 - \frac{f(N)}{N\alpha} (1 - RM) \right) = \lim_{a \to 0} \left( 1 - \lim_{a \to 0} \frac{f(N)}{N\alpha} (1 - RM) \right) = \lim_{a \to 0} N\alpha
\]

where the second equality was written using the facts that \( \lim_{a \to 0} f(N)^{a} = M \) and \( \lim_{a \to 0} (1-\varepsilon) = R \) (since the sequence is capacity achieving). The third equality is true because \( \lim_{a \to 0} N\alpha > 0 \) based on Lemma 6-1. The last equality is true based on Lemma 6-5.

Therefore for the first limit in right hand side of the last inequality of (C-16), we have:

\[
\lim_{a \to 0} \alpha \log \left( N\alpha - f(N) + (1-\varepsilon) f(N)^{a+1} \right) = \lim_{a \to 0} \alpha \log N\alpha = -\log R. \quad (C-19)
\]

where the first equality was written using (C-18) and the second equality was written using Lemma 6-4 and the fact that \( \lim_{a \to 0} \alpha \log \alpha = 0 \)

From (C-17) and (C-19) we conclude that:

\[
\lim_{a \to 0} \sup D^{-1}_c \log \left( 1 - \frac{\varepsilon}{1-R} \right) = -\log(M). \quad (C-20)
\]

With similar arguments, one can see that (C-20) is also valid for GC ensemble sequences.

Inverting both sides of (C-20) and multiplying by -1, we obtain

\[
\lim_{a \to 0} \sup (-D_c) / \log \left( 1 - \frac{\varepsilon}{1-R} \right) = 1/\log M.
\]

For the case of GR sequences, \(-\log R\) is a positive constant. Therefore

\[
\lim_{a \to 0} \sup \frac{D_c \log(R)}{\log \left( 1 - \frac{\varepsilon}{1-R} \right)} = -\log R / \log M = \mu.
\]

This proves the theorem for GR sequences.

For the case of GC sequences, \(-\log(1-\varepsilon)\) is a positive constant. Therefore using (C-20), we have
\[
\lim_{a \to 0} \text{SupDc} \log(1 - \varepsilon) / \log \left(1 - \frac{\varepsilon}{1 - R}ight) = -\log(1 - \varepsilon) / \log M = \mu.
\]

This proves the theorem for GC sequences. ■

**Proof of Corollary 6-1:**

First note that the bounds on \( \beta \) are necessary to ensure that \( f(N) < N \) and the function \( f \) is strictly increasing. For GC and GR ensemble sequences we have:

\[
M = \lim_{a \to 0} f(N)^{\alpha} = \lim_{a \to 0} (aN^\beta + b)^{\alpha} = \lim_{a \to 0} a^{\alpha} N^{\alpha \beta} = (\lim_{a \to 0} N^{\alpha})^\beta
\]

where the third equality is true because \( b \) can be neglected as \( aN^\beta \) tend to infinity.

Now based on Lemma 6-4 for GC ensemble sequences we have \( (\lim_{a \to 0} N^{\alpha})^\beta = 1/(1 - \varepsilon)^\beta \)

and therefore: \( \mu = -\log(1 - \varepsilon) / \log(M) = \frac{1}{\beta} \)

For GR ensemble sequences based on Lemma 6-4 we have \( (\lim_{a \to 0} N^{\alpha})^\beta = 1/R^\beta \) and

therefore: \( \mu = -\log(R) / \log(M) = \frac{1}{\beta} \).

This completes the proof. ■

**Proof of Theorem 6-5:**

We first consider GR ensemble sequences. Using, \( \lim_{a \to 0} \frac{R^{-1/\alpha}}{f(N)} = Q \), we can write

\[
\lim_{a \to 0} \frac{R^{-1}}{f(N)^{\alpha}} = \lim_{a \to 0} \left( \frac{R^{-1/\alpha}}{f(N)} \right)^{\alpha} = \lim_{a \to 0} Q^{\alpha} = 1.
\]

Therefore we conclude that
\[
\lim_{n \to 0} f(N)^\alpha = \frac{1}{R}
\]  

(C-21)

This means that \( \mu = 1 \) which is consistent with the results of [22]. From Lemma 6-3 we have:

\[
\lim_{\alpha \to 0} \sup \Delta(\lambda_n, \rho_n) = \lim_{n \to 0} \sup R^{-e} (1 - \frac{e}{1-R})
\]

\[
= \lim_{n \to 0} R^{-[(1-1/\alpha)/\alpha]} \left( \frac{1}{f(N)^{\alpha+1}} - \frac{1}{\alpha N \left( \frac{1}{f(N)^{\alpha}} - R \right)} \right)
\]

We therefore have:

\[
\lim_{\alpha \to 0} \sup \Delta(\lambda_n, \rho_n) = \lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{f(N)^{\alpha+1}} - \lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{\alpha N \left( \frac{1}{f(N)^{\alpha}} - R \right)} = \]

\[
\lim_{\alpha \to 0} \frac{R^{-1}}{f(N)^{\alpha}} \lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{f(N)} - \lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{N} \lim_{\alpha \to 0} \frac{1}{\alpha} \left( \frac{1}{f(N)^{\alpha}} - R \right) = 1 \times Q - R^{-1} e^f \vartheta ,
\]

where in the last equation, we have replaced the first limit based on (C-21), the second and fourth limits based on the theorem assumptions, and the third limit based on Lemma 6-3.

We therefore have

\[
\Delta = Q - R^{-1} e^f \vartheta .
\]

The proof for GC ensemble sequences is similar and thus omitted.

**Proof of Corollary 6-2:**

It is enough to compute the values of \( \vartheta \) and \( Q \) (assuming that the limits in Theorem 6-5 exist).

For GR sequences, we have:

\[
\lim_{\alpha \to 0} \frac{R^{-1/\alpha}}{f(N)} = \lim_{\alpha \to 0} R^{-1/\alpha} \frac{1}{a N^\beta + b} = \lim_{\alpha \to 0} R^{-1/\alpha} \frac{1}{a N^\beta} = \lim_{\alpha \to 0} R^{-1/\alpha} \frac{1}{N^{\beta+1}} = \frac{1}{a} \lim_{\alpha \to 0} \frac{1}{N^{\beta+1}} ,
\]

\]
where for the last equality, we have used Lemma 6-3. It is easy to see that if $\beta = 1$, the
limit exists we compute $Q = \frac{1}{a}e^{r}$. To compute $\partial$ for $\beta = 1$, we have

$$
\partial = \lim_{a \to 0} \frac{1}{\alpha} \left( \frac{1}{f(N)^{a}} - R \right) = \lim_{a \to 0} \frac{1}{\alpha} \left( \frac{1}{(aN + b)^{a}} - R \right)
$$

where the third equality is true because $b$ can be neglected as $N$ tends to infinity and the
last equation can be obtained using $L'Hôpital's$ rule. Using Theorem 6-5, we therefore
have

$$
\Delta = e^{r} \left( \frac{1}{a} - \ln a \right).
$$

This completes the proof for $GR$ sequences. The proof for $GC$ sequences is similar.
 References


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.