"In the name of God, Most Gracious, Most Merciful"

University of Alberta

LDPC Code Design for Nonuniform Channels

by

Ali Sanaei ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Communications

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2008

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-47404-4
Our file Notre référence
ISBN: 978-0-494-47404-4

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.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

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.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **LDPC Code Design for Nonuniform Channels** submitted by Ali Sanaei in partial fulfillment of the requirements for the degree of **Master of Science** in *Communications.*

Dr. Masoud Ardakani
Supervisor

Dr. Vincent Gaudet
Examiner

Dr. Mahammad Salavatipour
External Examiner

Date: ______________

To my parents...

Abstract

Notwithstanding the fact that many communication channels are best modeled as nonuniform channels, i.e., a set of parallel subchannels, most of the literature on channeling coding is focused on the much simpler model of uniform channels. The first part of this thesis presents new insight in LDPC code design considerations for nonuniform channels. In the second part, a nonuniform channel model comprising subchannels of equal capacity is used to devise a method for designing universal LDPC codes. Universal LDPC codes are designed given only the capacity of the channel and have satisfactory performance on all channels with the given capacity. This means the universal code is inoculated against those changes in the statistical behavior of the channel that do not affect capacity. Moreover, universal codes can be indexed according to only the capacity for which they are designed.

Acknowledgements

During my MSc study, I was blessed with the support of many exceptional individuals. I can never thank them enough, nor can I name all of them here, but I would like to use this opportunity to humbly express my gratitude toward this wonderful people.

I am fortunate beyond my imagination to have been benefitted immensely from the advice and encouragement of my affable mentor, Dr. Masoud Ardakani. I cannot even begin to thank him for his care, his support, and the influence that he has had on me.

I am grateful to the members of my committee, Dr. Gaudet and Dr. Salavatipour for their time and advice. I am also grateful to the University of Alberta for the opportunity to continue my graduate studies and for being a great host.

I would like to thank all my friends at iCORE Wireless Communications Lab., for many fruitful discussions and for making my study more enjoyable. My special thanks go to Raman Yazdani, Mahdi Ramezani, Moslem Noori, and Mahdi Hajiaghayi, for this work would have not been possible without their help.

Even though I have been living thousands of miles away from my family, I could feel the warmth of their support. I wish to express my eternal gratitude to them for their everlasting encouragement in all aspects of my life.

Contents

List of Figures

List of Tables

List of Abbreviations

List of Symbols

Symbol	Definition	First Use
C	channel capacity	1
$\mathbf{G}$	generator matrix	3
$\mathbf{H}$	parity-check matrix	3
R	code rate	3
ϵ	erasure rate in BEC; crossover probability in BSC	9
σ_n^2	variance of n	9
$f(.)$	LLRPDF	10
$g(.)$	PEPDF	11
λ	variable side edge-based degree distribution	14
ρ	check side edge-based degree distribution	14
SNR*	SNR decoding threshold	17
Λ	variable degree distribution for allotted LDPC	20
Var(x)	variance of x	29
E_b/N_0	power per information bit/noise power	36
H	entropy	55

Chapter 1

Introduction

Part of the appeal of digital communication is due to the possibility of having arbitrarily reliable transmission over a noisy channel through the use of error control coding. One of the most flexible and powerful classes of error correcting codes is the class of low-density parity-check (LDPC) codes which have attracted considerable attention during the last decade. This work is concerned with the design of LDPC codes for nonuniform channels and the design of channel-independent (or universal) LDPC codes.

This chapter begins with a brief overview of digital communication and channel coding. Then, LDPC codes and some of the preliminary concepts are introduced. After that, the motivation for the research and a brief historical review are discussed.

1.1 Basic Concepts

A block diagram of the essential sections in a digital communication system is shown in Fig. 1.1. The goal is to reliably transmit information over a noisy channel. Shannon, in his seminal work published in 1948, founded the field of information theory [2]. One of his most interesting results was that arbitrarily reliable communication over noisy channels is possible up to a specific rate of information bits per channel use called the *channel capacity* (C).

In the diagram, there is a data source producing a sequence of binary numbers entering the system. Then, the source encoder removes the redundancy, thus compresses the data. At this point, the channel decoder adds redundancy to the information in order to allow the receiver to compensate for the damage that is caused

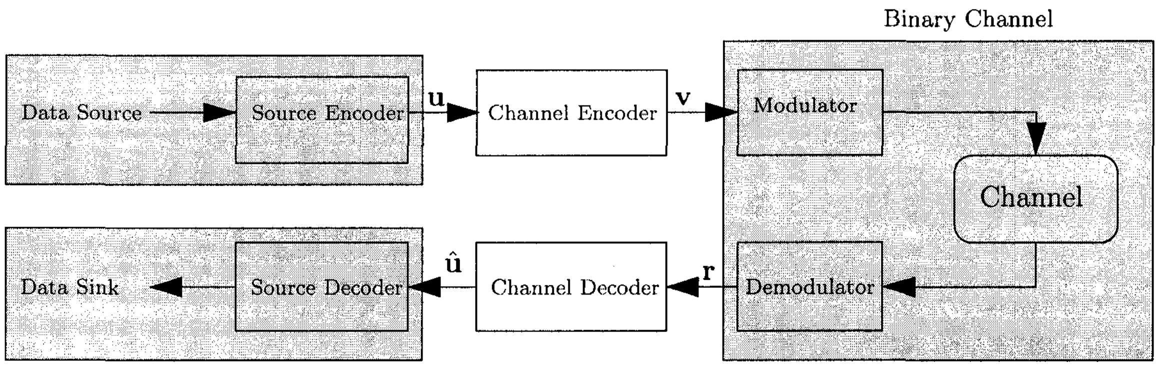


Figure 1.1: Block diagram of a digital communication system.

by the noisy channel. This means the channel coding should be chosen with regard to the statistical behaviour of the channel. The output is still a binary sequence which is passed to the digital modulator to be translated into a signal to be sent over the channel. Channel coding is the focus of this work and is meant by coding, unless otherwise noted.

The physical medium is called the communication channel. It might be, for instance, the atmosphere or an optical fibre. The channel usually changes the original signal to some extent. Hence, the received signal is noisy and corrupt.

At the receiver, the demodulator translates the received corrupt signals into a sequence of likelihoods. These likelihoods show the probability of being zero or one for each bit of the sequence. The decoder uses these likelihoods as well as the dependency of them to recover the original sequence. Finally, the source decoder decompresses the decoded sequence. Communication is successful when the original data is retrieved at the end.

1.1.1 Channel Coding

Since 1948, there has been continuous effort to find practical solutions that allow for utilizing the channel capacity. Certain capacity achieving methods have been known for a long time, but they all demand impractical complexity [3]. Over the years, a host of methods have been developed.

A binary block code is a one-to-one map from $[0,1]^k$ space to a subspace of $[0,1]^n$ $(n > k)$. Notice that everything is in $GF(2)$ so all the summations, for example, are to be performed in modulo 2. Information to be transmitted is parsed into k-bit vectors. Each k-bit information vector is then mapped, or *encoded*, to an n-bit coded vector, called a codeword. The rate of the code is defined as $R = \frac{k}{n}$, where n is called the code length.

The subspace of the code can be defined by a $k \times n$ matrix of rank k, called the generator matrix (shown by $\mathbf{G}_{k \times n}$). Using the generator matrix, an information vector, $\mathbf{u}_{1 \times k}$, can be encoded as $\mathbf{v}_{1 \times n} = \mathbf{u}\ \mathbf{G}$. A linear block code is fully described by its generator matrix. Notice that the generator matrix, however, is not unique.

A linear block code can also be specified by its null space, i.e., the set of vectors that satisfy $\mathbf{G}\mathbf{x}^T_{1 \times n} = 0$. The null space of a code is described by another matrix (which is not unique) called the parity-check matrix (shown by $\mathbf{H}_{(n-k) \times n}$), such that every binary vector $\mathbf{v}$ is a codeword in the code space if and only if,

$$\mathbf{H}\ \mathbf{v}^T = \mathbf{0}_{(n-k) \times 1}. \qquad (1.1)$$

Using this relation, it is not hard to determine whether a received vector is a codeword of the given code space or not. But the problem is that most of the times, the received vector is indeed a corrupted vector, and we want to remove the errors, i.e., to choose the codeword which was sent most likely. Decoding can be defined as making the best decision for the transmitted sequence, given the received (noisy) sequence.

Unfortunately, in order to obtain a better performance, one ought to increase the length of the code. For this reason, we are interested in optimal, or satisfactorily close to optimal, decoding schemes for which the decoding complexity is linear with the block length, such that an increase of the code length up to a reasonable length ($n = 10^5$, for today's applications) does not result in unreasonable increase in decoding complexity. For most of the classic coding techniques, the encoding/decoding computational complexity is not a linear function of the code length.

In the beginning of the 1990's, using convolutional codes, it was possible to approach the capacity by a gap of typically $4 - 5$dB. Better performance was impractically complex until turbo codes were discovered in 1993 [4]. An important breakthrough in turbo codes was the utilization of a class of suboptimal decoding rules, namely, iterative message passing algorithms. Using such algorithms to de-

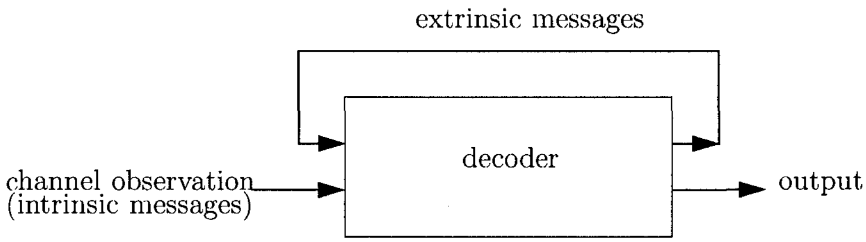


Figure 1.2: Iterative decoding is based on extracting the dependence within the code structure to find a better estimate for the sent codeword.

code turbo codes, one can obtain very close to capacity performance with a practical decoding complexity. Fig. 1.2 illustrates the idea of iterative decoding.

Soon after the advent of turbo codes, a more general category of codes were rediscovered, i.e., codes defined on graphs, that shared the remarkable features of turbo codes. That is, it was possible to find codes defined on graphs that could perform very close to capacity under iterative message passing algorithms. The search for codes defined on graphs which could perform extremely close to capacity led to the rediscovery of LDPC codes in 1996.

1.1.2 LDPC Codes

When the parity-check matrix (H) of a linear block code is sparse [1], the code is called LDPC. LDPC codes were invented by Robert Gallagar [5] in his PhD thesis in 1963 but had been largely ignored for a long time. At that time, these codes were considered too complex to be implemented for practical purposes. In 1996, LDPC codes were rediscovered by two independent groups, both influenced by the research on turbo codes [6, 7].

Soon it was shown that some LDPC codes are superior to the best turbo codes of the same length and can approach the Shannon capacity by a fraction of a decibel in practice [8].

LDPC codes became attractive also because of the simplicity of their graphical representation, based on Tanner graphs [9]. This simplicity allows for accurate analysis and design of LDPC codes. Moreover, LDPC codes are very flexible and therefore can be optimized for the specific channel over which the code will be used.

[1] A sparse matrix is one which is populated primarily with zeros.

1.2 Motivation

There are still a host of challenging problems with regard to the application of LDPC codes in realistic scenarios. For example, problems such as analysis of short-length LDPC codes, design of LDPC codes for wireless channels, new encoding methods, etc. are still appearing on the titles of the papers in the field.

This work is concerned with two related problems. The first problem is the design of LDPC codes for parallel channels. The second problem is LDPC codes that can be designed independent of the channel, given only the capacity.

1.2.1 Code Design for Parallel Channels

Many communication systems can be modeled as a group of parallel channels. For example, discrete multi-tone (DMT) systems are made up of different frequency tones that each has its own signal to noise ratio (SNR) . When the statistical behavior of the parallel subchannels is different, the system is called nonuniform. The nonuniformity makes the design of efficient channel coding solutions for such channels challenging, especially if we require all the tones to be coded and decoded in a single code.

Using a single code for protecting the symbols in all frequency tones has many practical benefits. For example, it allows for employing codes with long block-length for a practical buffer delay. In addition, using a single code reduces the software and hardware complexity of the whole system, compared to, say, one code for each subchannel. In the past few years, some of the modern coding techniques such as LDPC coding and turbo coding have been proposed for DMT systems [10, 11]. The motivation for using these codes has been their good performance with practical complexity.

It could be shown that taking into account the nonuniformity of the channel at the transmitter side does not change the capacity of the channel. That is, if we have a well-designed code for the channel, the channel could be used without the prior knowledge at the transmitter side. This means, having well designed codes, we can omit the feedback link. This is an important reduction in the cost.

The problem of efficient code design under different scenarios with regard to the availability of channel state information (CSI) at the transmitter and the receiver is the focus of the current work in this Chapter.

1.2.2 Design of Universal LDPC Codes

Quite related to the problem of code design for parallel channels, there lies a more challenging problem: without exact specification of the channel, is it possible to design codes that perform near capacity?

All of the LDPC code design algorithms need exact specification of the channel to give an appropriate code for that channel. In contrast, we rarely have such exact knowledge of the channel state (especially in the transmitter side). This fact becomes more important in wireless channels where the channel shows dramatic variations. Today's systems usually use a fixed channel code, where they work below their limit when channel has good quality and become useless when channel is in a deep fade.

It has been observed by many researchers [12–14] that codes which are designed for one channel model have a generally good performance on some other channel models provided the channels have equal capacities. So, if the only given specification of the channel is its capacity, is it possible to design a code that performs close to capacity on the channel?

Finding universal codes will allow the transmitter to use an error-correcting code only based on the channel capacity and not the channel state information. This will simplify the design of communication systems because the receiver usually has a good estimate of the channel and can compute the channel capacity and feedback this number to the transmitter. It is very inefficient to feedback the detailed channel state information to the transmitter, whereas one single number can represent the channel capacity. More interestingly, if universal codes that efficiently work on all equal-capacity channels exist, one can produce a library of codes optimized for different capacities and all systems can use codes of this library.

1.3 Thesis Outline

This thesis is organized as follows. Chapter 2 is a brief review on the preliminary concepts regarding design and analysis of LDPC codes.

Chapter 3 is dedicated to the LDPC code design considerations for nonuniform channels. Different scenarios regarding the presence or absence of the CSI are investigated. A new framework for code design when both sides have the CSI is proposed. As a practical example, code design for a power-line communication is considered.

In Chapter 4 the problem of universal LDPC code design is investigated. A method is suggested that allows for decomposing a channel into basis subchannels of equal capacity. Using this new decomposition method, a technique is proposed that allows for design of universal LDPC codes.

Finally, Chapter 5 concludes the thesis with a summary of the contributions of this work and suggestions for future research.

Chapter 2

Design and Analysis of LDPC Codes

In this chapter the necessary background on LDPC codes is discussed. Our channel model is introduced in Section 2.1, and three different methods of channel description are presented in Section 2.2. Section 2.3 is a brief review of the analysis of LDPC codes. Then, in Section 2.4 allotted LDPC codes which are LDPC codes especially modified for use on nonuniform channels are introduced.

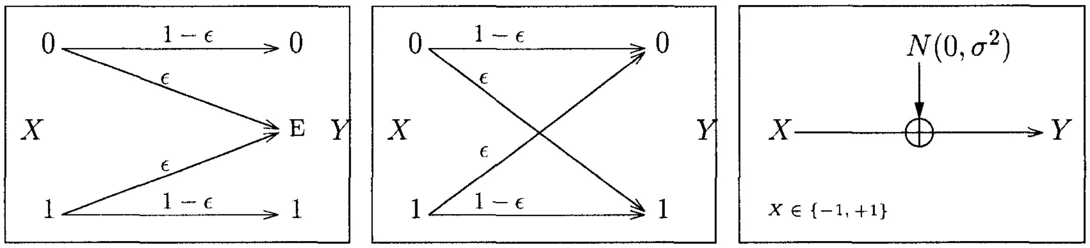
2.1 Channel Model

Channel coding is often concerned with a binary-input channel. Such a channel can have nonbinary output. If X is the input to a *memoryless* channel and Y is the output, the channel can be described by its conditional probabilities, i.e., $p(Y|X)$. When Y belongs to a discrete alphabet, $p(Y|X)$ can be shown by its transmission matrix, $M = [p_{i,j}]$, where $p_{i,j} = p(Y_j|X_i)$. When all of the rows of the transmission matrix are permutations of each other, and also all the columns are permutations of each other, the channel is called symmetric [3].

Most of the channels in this work are not symmetric with the above definition but enjoy some level of symmetry called symmetric-output. A binary-input symmetric-output (BISO) channel is one that abides by

$$p(Y|X = 0) = p(-Y|X = 1), \tag{2.1}$$

where $X \in \{0, 1\}$, $Y \in \mathbb{R}$.



(a) BEC (b) BSC (c) BIAWGN

Figure 2.1: Three binary-input channel models.

The following examples define three of the channels frequently referred to throughout this work. These channels are the binary erasure channel (BEC) , binary symmetric channel (BSC) , and binary-input additive white Gaussian (BIAWGN) channel. These channels are depicted in Fig. 2.1.

A BEC is a channel that either transmits a bit correctly (with probability $1-\epsilon$) or erases the bit, by giving the receiver an erasure symbol. The following transmission matrix describes a BEC.

$$M_{\text{BEC}} = \begin{bmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{bmatrix}.$$

A BSC is similar but instead of losing bits, flips (inverses) them with probability ϵ. The following transmission matrix describes a BSC.

$$M_{\text{BSC}} = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}.$$

A binary-input channel might have a continuous alphabet at the receiver side. A BIAWGN transmits $X \in \{+1, -1\}$ and the signal is corrupted by an additive white Gaussian noise. This means that the received signal (Y) can have any real value. The following equation describes a BIAWGN.

$$y = ax + n,$$

where a is the channel gain and n is the additive white Gaussian noise. For the sake of simplicity, it is assumed that $a = 1$. So, the noise power is expressed in terms of the SNR as

$$\sigma_n^2 = \frac{1}{\text{SNR}}.$$

2.2 Channel Description

A BISO channel can be completely described by the probability density function (pdf) of its log-likelihood ratio (LLR) . The pdf of the LLR (LLRPDF) is obtained by assuming that the all-zero codeword is transmitted. When bit X is sent and Y is received, the LLR (denoted by m) is defined as

$$m = \log \frac{p(X=0|Y)}{p(X=1|Y)}.$$

Therefore, there is a one-to-one correspondence between the pdf of m, $f(m)$, and the channel. The capacity of the channel can be obtained by calculating the mutual information between X and Y, which is a function of $f(m)$. We denote the mutual information of $f(m)$ by $\mathcal{I}(f(m))$, which can be calculated as[1] [15]

$$\mathcal{I}(f(m)) = 1 - \mathrm{E}_m\{\log_2(1+e^{-m})\} = 1 - \int_{-\infty}^{+\infty} f(m)\log_2(1+e^{-m})dm. \tag{2.2}$$

Thus, the maximum achievable rate on this channel, i.e., the capacity of this channel, $C = \mathcal{I}(f(m))$.

Although every BISO channel has a unique LLRPDF, not every pdf represents a BISO channel. To be more specific, a pdf has to satisfy the symmetry condition [15], i.e., $f(-m) = e^{-m}f(m)$ in order to be a valid LLRPDF of a BISO channel. The fact that not every pdf is a valid LLRPDF imposes some difficulties in Chapter 4 where channel decomposition is performed.

Another channel representation which does not involve the above constraint and allows every pdf to represent a BISO channel is based on probability of error [16],

$$p = \min\{p(X=1|Y), 1-p(X=1|Y)\}. \tag{2.3}$$

It follows that $p \in [0, \frac{1}{2}]$. We represent the pdf of probability of error (PEPDF) with $g(p)$.

For example, a BSC with crossover probability of ϵ can be represented with its LLRPDF as

$$f(m) = (1-\epsilon)\delta(m - \log\frac{1-\epsilon}{\epsilon}) + \epsilon\delta(m + \log\frac{1-\epsilon}{\epsilon}),$$

[1]A more general form of this equation will be proven in Chapter 3.

which consists of two delta functions satisfying the symmetry condition. The channel can also be represented by $g(p) = \delta(p - \epsilon)$.

As another example consider

$$g(p) = \gamma\delta(p - \epsilon_1) + (1 - \gamma)\delta(p - \epsilon_2).$$

This represents a channel which consists of two BSCs, one with crossover probability of ϵ_1 and another one with crossover probability of ϵ_2. The first channel passes γ fraction of transmitted bits and the second one $1 - \gamma$ fraction of bits. Similarly, for the continuous cases, the PEPDF can be obtained. For example, the PEPDF of a Gaussian channel is [16]

$$g(p) = \frac{\sigma}{p(1-p)^2\sqrt{8\pi}} e^{-\frac{\sigma^2}{8}(\log\frac{1-p}{p} - \frac{2}{\sigma^2})^2}.$$

From the definition and these examples, it should be clear that unlike LLR-PDF, PEPDF representation allows every pdf defined over $p \in [0, \frac{1}{2}]$ to represent a BISO channel. In fact, this representation decomposes a BISO channel into a convex combination of BSCs of different crossover probabilities. The fraction of bits passing through each BSC is captured in PEPDF. The three channel representations presented in this section are interchangeable when the input is uniformly distributed [16].

In Chapter 4, we will propose a new representation for the channel which is obtained from the PEPDF, but describes the channel as a convex combination of identical-capacity basis channels.

2.3 Analysis of LDPC Codes

LDPC codes are in part appealing because of the powerful analytic methods that code designers have at their disposal. Most of these methods are based on a graphical representation of LDPC codes.

2.3.1 Graphical Representation

The idea of using bipartite graphs to simplify design and analysis of linear codes goes back to Tanner's work in 1981 [9], and hence the term Tanner graph means the bipartite graph that represents a linear code.

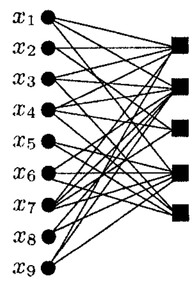


Figure 2.2: Factor graph and parity-check equations of a short linear block code defined in (2.4).

A graph is said to be bipartite when the nodes can be partitioned into two disjoint classes. Each edge in the bipartite graph should connect a nodes from one class to a node from the other class. A well-known example of bipartite graphs in probabilistic reasoning is a factor graph. In a factor graph, one class of nodes are function nodes and the other class are variable nodes. Factor graphs are very efficient tools for obtaining the marginals of a multivariate function when the function can be factorized into simpler functions [17, 18].

It was stated that a set of parity-check equations (1.1) can determine whether or not a vector of the appropriate size belongs to a codebook, i.e., is it a codeword or not? This set of equations can be used to make a factor graph in which each function node, hereinafter called *check node*, illustrates one of the equations.

Consider the simple LDPC code of length 9 and rate $R = \frac{4}{9}$ described by the following parity-check matrix:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{2.4}$$

Fig. 2.2 demonstrates the factor graph and parity-check equations of this code as well as the corresponding factor graph. The variable nodes and the check nodes are located in the left side and right side of the graph, respectively. If all the check nodes, i.e., the equations, are satisfied, the vector that the variable nodes show is a codeword.

Notice that the number of edges in the factor-graph of a code is identical to the number of 1's in the parity-check matrix of the code. The number of edges in the corresponding factor graph of LDPC codes increases linearly with the code length. Hence, the computational complexity of the message-passing algorithm for LDPC codes increases linearly with the length of the code. In other words, the complexity per bit of LDPC code remains almost constant and does not depend on the code length.

2.3.2 Code Structure

It is shown that the properties of LDPC codes with large code length are mostly determined by the length of the code and the distribution of the degrees of the two classes of nodes [19]. From the point of view of code structure, LDPC codes can be classifies into two categories: regular and irregular.

Whereas irregular codes can have variable nodes (and check nodes) of different degrees, regular codes have variable nodes of a fixed degree (d_v) and check nodes of a fixed degree (d_c). If the code length is n, the factor graph of the code has n variable nodes, and $(1-R)n$ check nodes, where R is the rate of the code. Denoting the number of edges in the graph with E, it follows that

$$E = nd_v = (1-R)nd_c.$$

Then rate of the code can be obtained as

$$R = 1 - \frac{E/d_c}{E/d_v} = 1 - \frac{d_v}{d_c}.$$

When n is large, almost all the regular LDPC codes with variable degree d_v, check degree d_c and length n behave very similarly [19]. So, for asymptotically long LDPC codes, instead of specifying a particular incident, the ensemble of LDPC codes of a particular d_v and d_c, is often specified. This ensemble is referred to as the ensemble of (d_v, d_c) regular LDPC codes.

Compared to many channel coding schemes, regular LDPC codes perform closer to capacity. Nevertheless, optimized irregular LDPC codes outperform regular ones [20]. Irregular LDPC codes are usually characterized by their edge-perspective right and left degree distributions. The variable side edge-based degree distribution (also known as the left degree distribution) is usually denoted by $\lambda = [\lambda_1, \lambda_2, \dots \lambda_{d_v}]$, where λ_i is the fraction of edges in the factor graph of the code incident to variable nodes of degree i. Also, d_v denotes the maximum variable node degree in the code. Similarly, the check side (or right) degree distribution is denoted by $\rho = [\rho_1, \rho_2, \dots, \rho_{d_c}]$, where d_c denotes the maximum check node degree in the code.

Consider an irregular LDPC code with left and right degree distributions λ and ρ and length n. Assume that the number of edges in the graph is E. There are $E\lambda_i$ edges connected to variable nodes of degree i. It follows that there are $E\lambda_i/i$ variable nodes of degree i. So,

$$n = \sum_i E\frac{\lambda_i}{i}.$$

Similarly, since the number of check nodes is $(1-R)n$, it follows that

$$(1-R)n = \sum_i E\frac{\rho_i}{i}.$$

Therefore, given λ and ρ, the rate of the code can be obtained as

$$R = 1 - \frac{\sum_{i=1}^{d_c} \rho_i/i}{\sum_{i=1}^{d_v} \lambda_i/i}.$$

Similar to regular LDPC codes, the performance of asymptotically long irregular LDPC codes on a given channel is determined by λ and ρ [8]. Hence, to design an LDPC code is usually tantamount to finding the left and right degree distributions that assure the required performance on the given channel.

2.3.3 Sum-product Algorithm

There are a number of message-passing decoding algorithms based on the factor graph representation of linear codes. Sum-product is an algorithm for approximate inference on graphical models. It has been shown that if the graph is a tree (i.e., without cycles) the exact algorithm converges to a fixed point which is the optimal solution. For loopy graphs (graphs with cycles), however, the algorithm is neither guaranteed to converge to a fixed point, nor is the result guaranteed to be the correct answer. There has been ongoing research in finding a practical set of necessary conditions that assure the convergence of the algorithm to an optimal answer on loopy graphs [21].

Although a suboptimal algorithm in general, sum-product algorithm can perform surprisingly close to a maximum likelihood (ML) decoder in most cases. Close to optimal performance, in tandem with the fact that the complexity of the algorithm increases linearly with the length of the code, has made sum-product algorithm the most attractive decoding algorithm in the context of LDPC codes.

Assume that the length of the code is n and the code rate is R. So, there are n variable nodes, $\{v_1, ..., v_n\}$ and $(1 - R)n$ check nodes, $\{c_1, ..., c_w\}$. The set of check nodes directly connected to each node, like v_j is denoted by $\mathcal{N}\{v_j\}$.

Variable nodes are initially loaded with the values observed from the channel. This value is the likelihood of having received 0 or 1. Here, it is assumed that variable nodes receive LLR values, and the values passed on edges are also assumed to be LLR. The initial LLRs are denoted as $m_{\mathrm{ch},i}$.

As mentioned before, sum-product is an iterative message-passing algorithm. At the initial stage, or iteration 0, each variable node passes its value to the neighboring check nodes. Thus, for each variable node v_i's output,

$$m^{(0)}_{v_i \to c_j} = m_{\mathrm{ch},i} \quad .$$

These messages are called intrinsic messages because they are observed from the channel and not affected by the constraints that the code should abide by.

The following iterations are comprised of two phases of computing and sending messages: from variable nodes to neighboring check nodes and vice versa. At any iteration, say iteration ℓ, first, each check node performs the following operation and passes the result to the neighboring variable nodes [18].

$$m^{(\ell)}_{c_j \to v_i} = 2\tanh^{-1}\left(\prod_{v_k \in \mathcal{N}(c_j) - v_i} \tanh\left(\frac{m^{(\ell-1)}_{v_k \to c_j}}{2} \right) \right) . \tag{2.5}$$

These messages are called extrinsic messages.

Then, the variable nodes perform the following equation and pass the result to their neighboring check nodes,

$$m^{(\ell)}_{v_i \to c_j} = m_{\mathrm{ch},i} + \sum_{c_k \in \mathcal{N}(v_i) - c_j} m^{(\ell)}_{c_k \to v_i} \quad . \tag{2.6}$$

Equations (2.5) and (2.6) constitute one iteration. When the algorithm is used for decoding, the iterative process of sum-product is performed up to a maximum number of iterations or until all the check nodes are satisfied. Each check node is said to be satisfied when its corresponding parity-check equation holds. In other words, a check node is satisfied when the message it sends to each of its neighboring variables has the same sign as the last message that it has received from that variable.

At iteration ℓ, if one is to decide for the value of the sent bits, they can be obtained by combining the intrinsic messages and the last set of extrinsic messages as follows,

$$v_i = \text{sign}\left(-m_{\text{ch},i} - \sum_{c_k \in \mathcal{N}(v_i)} m^{(\ell)}_{c_k \to v_i}\right) ,$$

where the sign function returns 0 when its argument is negative, returns 1 when its argument is positive, and returns either 0 or 1, equally likely, when its argument is zero.

There are a number of simplifications of the sum-product algorithm which suit particular practical purposes. For example, instead of LLR, hard information (zero or one) might be observed from the channel. Another example is the min-sum algorithm which simplifies the operation at check nodes. Further discussions in this regard can be found in [18].

2.3.4 Density Evolution

The rationale of iterative decoding is that the extrinsic messages usually improve, or *evolve*, iteration by iteration. So, tracking the evolution of the statistical features of the messages is a natural means for analyzing a given code. Density evolution has two main assumptions [19]: (1) The code length tends to infinity which also means the factor graph is a tree (2) the channel is BISO and the decoding algorithm is symmetric. Before introducing the method, these assumptions will be explained.

If the factor graph is cycle-free, i.e., a tree, at every iteration, the incoming messages are independent. When there are cycles present in the factor graph, the girth of the graph is defined as the length of the shortest cycle. If the girth of a factor graph is ψ, the graph from the point of view of each variable node is similar to a tree for at least the neighborhood of depth $\lfloor \frac{\psi}{2} \rfloor$. Therefore, the extrinsic messages that each variable node receives are independent from its intrinsic message for at least $\lfloor \frac{\psi}{2} \rfloor$ iterations.

For randomly constructed factor graphs, the ratio of short cycles to code length decreases as the length of the code increases. That is, for long codes, most variable nodes receive independent extrinsic messages for a healthy number of iterations. Therefore, in very long codes, the contribution of the cycles to the behavior of the codes is negligible, and vanishes asymptotically with the code length. Hence, the factor graph is assumed to be a tree in the asymptotic analysis of LDPC codes.

The second assumption is that the channel is BISO, defined in (2.3), and the update rules have symmetry condition. The update rules are said to be symmetric when, simply put, they treat negative and positive LLR values similarly. The update rules of sum-product algorithm are symmetric [19]. Under the assumption of BISO channel and symmetric update rules, the statistical behavior of a linear code does not depend on the actual codeword that is sent. That is, one can study the code assuming that the transmitted codeword is the all-zero codeword (present in all linear codes), and generalize the result.

Density evolution receives the LLRPDF of the messages received from the channel and tracks the LLRPDF of the extrinsic messages iteration by iteration. So, the update rules are the update rules of the decoding algorithm, but the messages are LLRPDFs instead of actual LLR values. Therefore, the intrinsic message is identical for all the variable nodes. Also, when the code length tends to infinity, for a randomly constructed code, the extrinsic messages at a given iteration depend only on the degree of the check and variable nodes. Therefore, it is sufficient to keep track of LLRPDFs only for check nodes and variable nodes of different degrees. The analytic formulation can be found in [19]

One of the most important parameters obtained by using density evolution is the code threshold. The code threshold for a given code and channel type is defined as the worst channel condition on which the code converges to zero error rate. For example, for a Gaussian channel the threshold of a code is usually given in the form of the worst possible SNR, shown by SNR*.

In most cases, density evolution is computationally complex. Discrete density evolution was devised to make computer implementations possible [22]. As the name suggests, the only difference is that the messages are quantized and hence, pmfs are used instead of pdfs. Numerical experiments show that with 11-bit quantization of

LLRs (2048 levels of quantization), the convergence of the code is approximated with less than 0.001 dB error [22], which is accurate enough for almost every purpose.

2.3.5 EXIT Charts

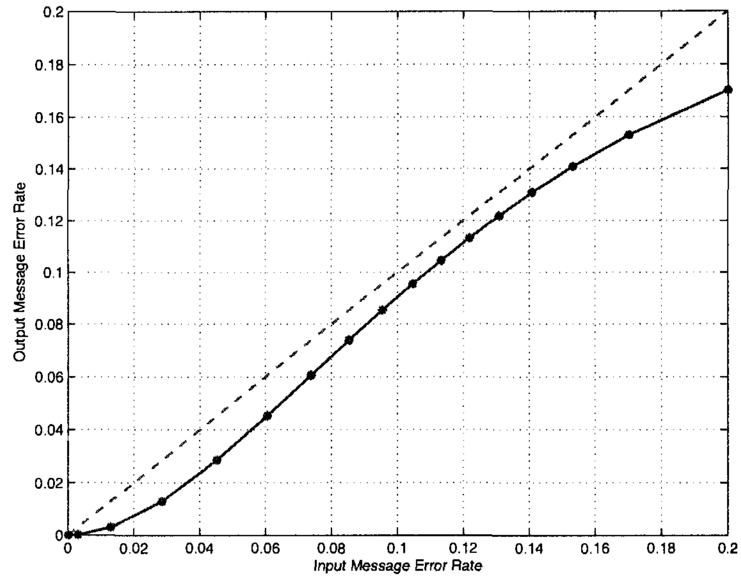
extrinsic information transfer (EXIT) chart is another method for analyzing the asymptotic behavior of a code under iterative decoding algorithms. EXIT chart analysis tracks the evolution of a specific parameter. There are various choices, but message error rate [23] and mutual information between the density and the all-zero codeword [24] are the more often used ones.

In a few cases where the LLRPDF of the intrinsic and extrinsic messages can be fully described by a single parameter (e.g., in BEC), EXIT chart analysis is exact and equivalent to density evolution. In most of the cases, however, it might not be an exact analysis. To have exact EXIT chart analysis, it is possible to perform density evolution but, for the purpose of analysis, extract a specific parameter such as message error probability at the end of each iteration. In this work, EXIT charts are exact and are obtained in this way.

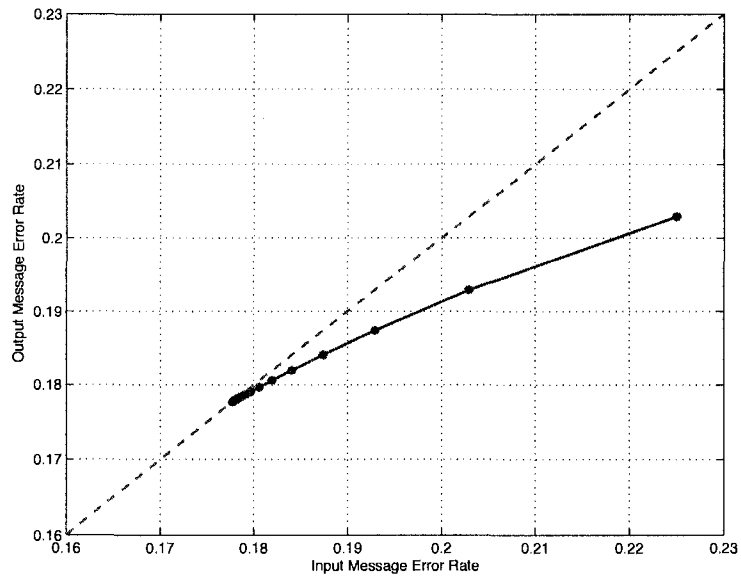
At each iteration of the density evolution, the error rate of the messages sent from variable nodes at the previous iteration and current iteration make a pair of message error rates like $(q^{(\ell-1)}, q^{(\ell)}) = (q_{\text{in}}, q_{\text{out}})$. After performing ℓ iterations of density evolution, one will have ℓ pairs which can be drawn as a trajectory, which is called an EXIT chart.

When message error rate is used as the representing parameter, the goal is to have a trajectory that finally reaches the zero error rate point. If at some point $q_{\text{in}} \leq q_{\text{out}}$, that is, if the trajectory crosses the $y = x$ line, it means the decoding does not converge to zero error rate. Fig. 2.3 shows the EXIT chart for (3,6) regular LDPC codes over two BECs with different erasure rates. It is evident that Fig. 2.3a indicates successful decoding whereas Fig. 2.3b suggests otherwise.

An EXIT chart is especially suitable for code design. EXIT charts are used in Chapter 3 to design allotted LDPC codes and in Chapter 4 to design universal LDPC codes.



(a) Successful convergence to zero error rate.



(b) Crossing the $y = x$ line indicates unsuccessful decoding.

Figure 2.3: Message error rate EXIT charts for (3,6) regular LDPC code over (a) BEC with erasure rate $\epsilon = 0.4$ and (b) BEC with erasure rate $\epsilon = 0.45$. It can beshown that the threshold is $\epsilon = 0.4294$.

2.4 Allotted LDPC Codes

The problem of designing a single LDPC code for a nonuniform channel has been previously studied in [25–27]. The gist of the idea is that different subsets of nodes, corresponding to different left degree distributions in a single Tanner graph, are connected to different subchannels. In other words, the code is *allotted* (in contrast with *conventional* codes) among subchannels.

To allow for this allotment, the definition of the left degree distribution has to be extended. This extension is done in a number of different ways. We incorporate the method suggested in [25], with a slight modification in notations as follows. The left degree distribution is broken into sub-degree distributions. So, instead of having a vector that represents left degree distributions, we have a matrix whose each row corresponds to one subchannel, i.e.,

$$\Lambda = [\lambda_{j,i}]_{K \times d_v} = [\Lambda^{(1)^T}, \Lambda^{(2)^T}, \dots, \Lambda^{(K)^T}]^T, \tag{2.7}$$

where, row j is the degree distribution for subchannel j, from an edge perspective, and K is the number of subchannels. Thus, row j can be viewed as $\Lambda^{(j)} = [\lambda_{j,1}, \dots, \lambda_{j,d_v}]$, where $\lambda_{j,i}$ denotes the portion of edges connected to degree i variable nodes that are transmitted through the jth subchannel. Check node distribution, similar to conventional LDPC codes, is defined by $\rho = [\rho_1, \rho_2, \dots, \rho_{d_c}]$.

Chapter 3

Nonuniform Channels

Non-unform channels are suitable models for many applications. They appear in networks where packets are sent through parallel routes with different qualities; in multi-input multi-output systems where signals experience different distortion levels in different channel pairs; in DMT and orthogonal frequency-division multiplexing (OFDM) channels where different frequency tones have unequal SNR, and even in storage media where different places on the disc have various qualities. This Chapter provides new insights in LDPC code design considerations for nonuniform channels[1].

3.1 Introduction

The problem of LDPC coding for nonuniform channels is studied in [25]. The solution is based on carefully assigning variable nodes of different degrees to different subchannels based on their qualities, i.e., using allotted LDPC codes. In allotted LDPC coding, the number of design parameters is significantly increased. Therefore, a search-based code design is inefficient. To overcome this problem, [25] suggests semi-regular codes, where no subchannel can be assigned to variable nodes of more than one degree. This limitation may result in suboptimal codes. Another solution studied in [30], is based on puncturing different parts of an LDPC code at different rates, which is by nature a suboptimal solution.

Allotted LDPC codes and their design have also been studied for OFDM systems and satisfactory performance is reported in [1] and [27]. [1] uses a linear criterion but relies on a Gaussian assumption [31], which, although proven accurate for the output

[1]Parts of this chapter has been previously published [28] or submitted for publication [29]

of variable nodes, is not a suitable assumption for the output of check nodes [31].

A more recent work on allotted LDPC codes derives the upper bound on the achievable rate of this coding method over binary-input symmetric-output parallel channels under maximum likelihood decoding [32].

By introducing some auxiliary integer parameters, it will be shown that the design of semi-regular codes can be solved via mixed integer linear programming. This solution is more efficient than an exhaustive search, but still more complex than the design of conventional codes. By relaxing the semi-regularity constraint in the design of allotted codes, the whole range of irregularity becomes available to all subchannels. This results in both improved codes and more efficient code design based on linear programming.

Having seemingly superior performance and a design method almost as efficient as that of the conventional LDPC codes, the only downside of allotted LDPC codes is that they require channel information at the transmitter side. Since channel information at the transmitter does not change the capacity [33], conventional codes in principle can perform as close to capacity as allotted ones. So, is there any benefit in using allotted codes? To answer to this question, assessing different LDPC coding approaches for nonuniform channels seems necessary.

We discuss the three different methods, namely semi-regular allotted LDPC coding, irregular allotted LDPC coding, and conventional LDPC coding in terms of code design complexity and performance. This comparison is performed in different situations with respect to availability of the channel state information (CSI) at the transmitter and/or the receiver. One important result is that under suboptimal decoding (e.g., hard decoding) or when maximum variable degree allowed in the code is small, allotted LDPC codes can significantly outperform conventional LDPC codes and should be the code of choice.

The sequel of this chapter is organized as follows. In Section 3.2, we briefly review some required background on conventional and allotted LDPC codes, as well as on nonuniform channels, and present the channel model. We investigate the design of semi-regular allotted LDPC codes in Section 3.3 and propose an iterative mixed integer linear programming (MILP) method. In Section 3.4 we investigate code design and code performance for the case where conventional codes must be used due to the absence of CSI at the transmitter side. Numerical results are partly presented in Section 3.5. As a practical case, code design for a power-line channel is considered in Section 3.6. Finally, Section 3.7 concludes the chapter.

3.2 Background and System Model

3.2.1 Nonuniform channels

While the literature on LDPC code design for different channel models abounds, usually the channel is assumed to be uniform. That is, different bits of the code go through different realizations of the channel, but channel parameters are similar for different bits. In many communication systems, however, the uniformity of the channel is an oversimplification, because the channel parameters that different parts of the code experience can be dramatically different. For example, in an OFDM system, different frequency tones may have vastly different SNRs—which is one of the reasons for using OFDM in the first place.

The model that we use is a channel comprising K parallel subchannels. Therefore, the symbols that go through different subchannels may experience different channel parameters. In our model, subchannels are statistically independent.

Subchannels can be described by their conditional probabilities. So, Subchannel j, $1 \leq j \leq K$, is defined by the pdf of receiving Y when X is transmitted, i.e., $p_j(Y|X)$. Nonuniform channels are usually made up of subchannels within a broad range of quality. In order to make use of the subchannels with good quality, higher order modulations should be employed. Suppose a constellation of Q_j signals is used for subchannel j so that it carries $\log_2 \ Q_j$ bits. We denote the fraction of the LDPC-coded bits that pass through subchannel j by γ_j, where $\sum_1^K \gamma_j = 1$.

We assume that all the subchannels use the same signalling (Q_j's are equal). The size of the constellations should be selected based on the affordable detection complexity and the typical quality of subchannels. Having various constellation sizes is called bit-loading. Our coding methods is applicable regardless of whether or not bit-loading is used.

Notice that using one constellation for all subchannels is a worst case scenario as it results in bit-channels with significantly different qualities. If a coding solution can handle such severe unequal qualities, it certainly handles cases that bit-loading is used and the constellation sizes are chosen according to the capacity of the subchannels. We also assume bit-interleaved coded modulation to avoid sequential decoding of bit-channels.

If Gray labeling is used, one can interleave these $\log_2 \; Q_j$ bits in a single code and ignore their dependencies. This way, sequential decoding of bits is avoided at a negligible performance degradation. This technique is called bit-interleaved coded modulation [34]. Bit-interleaved coded modulation for a specific subchannel results in *bit-channels* that have almost equal capacity. However, the bit-channel capacity may still significantly vary from one subchannel to another due to the various qualities of different subchannels.

Once the modulation is specified, without loss of generality, instead of dealing with K subchannels, we can use the corresponding bit-channels [35]. Thus, the channel can be modeled as N independent bit-channels with possibly different qualities. Both models are depicted in Fig.3.1. If the capacity of bit-channel j $(1 \leq j \leq N)$ is C_j, the total capacity is

$$C_{\text{total}} = \sum_{j=1}^{N} C_j. \tag{3.1}$$

It is also useful to define an *average capacity* as $C_{\text{avg}} = C_{\text{total}}/N$.

3.3 Efficient Design of Allotted LDPC Codes

When both the transmitter and the receiver know the CSI, it is possible to use allotted LDPC codes. The major difficulty about allotted codes is the design problem. The existing efficient code design procedures, developed for conventional codes, cannot be applied to allotted codes directly due to their different structure. Moreover, search based methods are particularly inefficient due to a considerable increase in the number of design parameters. In [25], a solution based on semi-regular codes has been suggested. The main reason for using semi-regular codes is to reduce the size of the search domain and hence allow for search based optimization.

Here, we first formulate the code design with the semi-regular constraint as a standard MILP. Later we will see that relaxing the semi-regular constraint both simplifies the code design problem and results in improved performance.

3.3.1 Semi-Regular Codes

The goal is to find the semi-regular LDPC code which achieves the highest rate of data transmission over this channel. Defining

$$\mathbf{U} = \left[1, \frac{1}{2}, ..., \frac{1}{d_v}\right], \tag{3.2}$$

it is easy to see that maximizing the code rate is equivalent to maximizing $\mathbf{1}_{1\times K}\Lambda\mathbf{U}^T$, where $\mathbf{1}_{1\times K}$ is a vector of ones with length K.

We denote LLR pdf of the messages that come from subchannel j with $f_j(m)$. At iteration ℓ, we denote the LLR pdf of the input messages to check nodes with $f^{(\ell)}(m)$. It is noteworthy that $f^{(\ell)}(m)$ is the same for all nodes as the structure of the LDPC code interleaves all the messages from the previous iteration (Fig. 3.2). We can also define an input message error rate for the iteration as

$$p^{(\ell)} = \int_{-\infty}^{0} f^{(\ell)}(m)dm.$$

Now, let us define $q_{j,i}^{(\ell)}$ as the message error rate at the output of degree i variable nodes that are assigned to subchannel j and $q^{(\ell)}$ as the average of $q_{j,i}^{(\ell)}$. The superscript ℓ shows the iteration number. Here, $q^{(\ell)}$ represents the message error rate at the output of this iterations, which is the input error rate for the next iteration, i.e., $p^{(\ell+1)}$. Assuming $f^{(\ell)}(m)$ as the LLR pdf at the input of the check nodes, the output of a degree i variable node assigned to subchannel j has the following LLR pdf [18]

$$g_{j,i}^{(\ell)}(m) = \bigotimes_{l=1}^{i-1} \mathrm{CHK}\big(f^{(\ell)}(m), \rho\big) \otimes f_j(m) \tag{3.3}$$

where $\mathrm{CHK}(f^{(\ell)}(m), \rho)$ is the LLR pdf after the check operation and $\otimes$ denotes convolution. Therefore,

$$q_{j,i}^{(\ell)} = \int_{-\infty}^{0} g_{j,i}^{(\ell)}(m)dm.$$

For the next iteration, the input LLR pdf to the check nodes can be written as

$$f^{(\ell+1)}(m) = \sum_i \sum_j \lambda_{j,i}\, g_{j,i}^{(\ell)}(m).$$

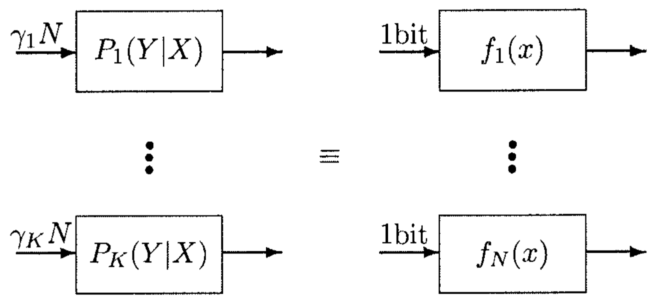


Figure 3.1: Two equivalent channel models. *Left:* The channel is a set of K parallel subchannels each carrying a certain fraction of N bits, all using constellation size q. *Right:* The channel is made up of N parallel bit-channels. The probability density function of LLRs received at bit-channel i is denoted by $f_i(x)$.

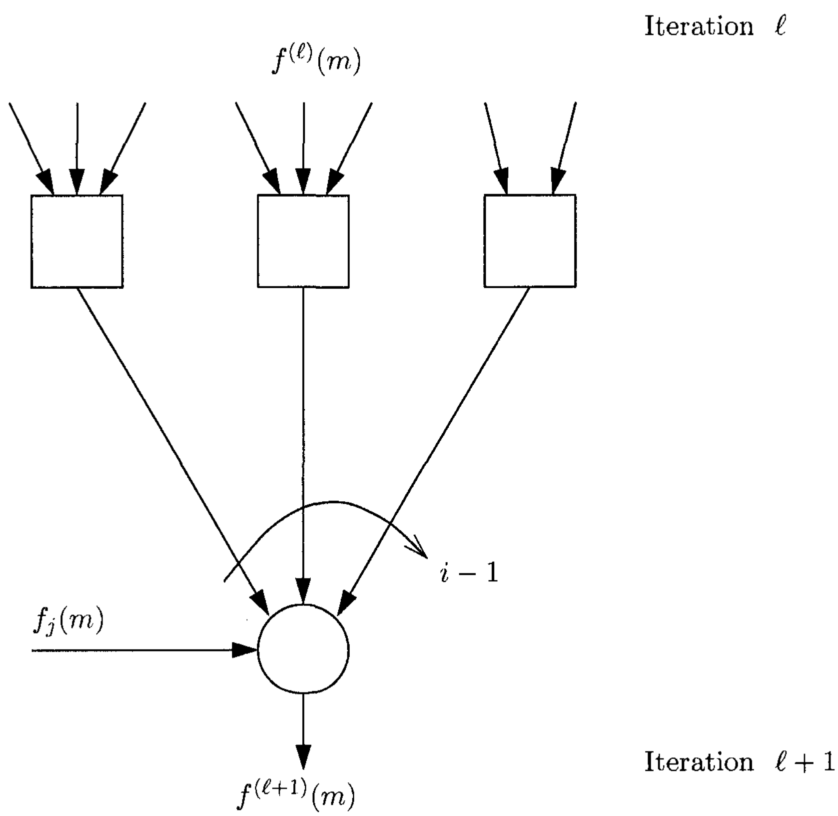


Figure 3.2: One iteration of density evolution.

Once density evolution is performed for N_i iterations, for each $p^{(\ell)}$, $1 \leq \ell \leq N_i$, we can define $\mathbf{q}(p^{(\ell)})$ as

$$\mathbf{q}(p^{(\ell)}) = [q_{j,i}^{(\ell)}]_{K \times d_v}.$$

Therefore, we have

$$p^{(\ell+1)} = q^{(\ell)} = \mathbf{1}_{1\times K}[\Lambda \bullet \mathbf{q}(p^{(\ell)})]\mathbf{1}_{d_v \times 1}, \tag{3.4}$$

where $\bullet$ denotes Hadamard product.

To ensure semi-regularity of the code, we have to enforce another constraint that lets only a single element in each $\Lambda^{(j)}$ be nonzero. While this is a non-linear constraint, we can circumvent the non-linearity by introducing a set of integer variables which have to be optimized along with Λ. We define $\Delta = [\delta_{j,i}]_{K \times d_v}$ as a matrix of binary integers such that

$$\forall i, j, \quad \lambda_{j,i} - \delta_{j,i} \leq 0$$

and $\sum_j \delta_{j,i} = 1$. The latter constraint ensures that only one $\delta_{j,i}$ in each row is equal to one and every other $\delta_{j,i}$ is zero. The former constraint ensures that only for the positions that $\delta_{j,i}$ are non-zero, $\lambda_{j,i}$ can have a non-zero weight.

So, the rate maximization problem can be written as follows. In this formulation, for two matrices (vectors) $A = [a_{i,j}]$ and $B = [b_{i,j}]$, $A \leq B$ means that $\forall i, j$, $a_{i,j} \leq b_{i,j}$.

$$\underset{\{\Lambda,\Delta\}}{\text{maximize}} \quad \mathbf{1}_{1\times K}\Lambda\mathbf{U}^T \tag{3.5}$$

subject to

$$\forall i, j, \quad \delta_{j,i} \in \{0, 1\} \tag{3.6}$$

$$\Lambda - \Delta \leq \mathbf{0}_{K\times d_v} \tag{3.7}$$

$$\mathbf{1}_{1\times d_v}\Delta^T = \mathbf{1}_{1\times K} \tag{3.8}$$

$$\Lambda \geq \mathbf{0}_{K\times d_v} \tag{3.9}$$

$$\mathbf{1}_{1\times K}\Lambda\mathbf{1}_{d_v\times 1} = 1 \tag{3.10}$$

$$\Lambda\mathbf{U}^T - \gamma\mathbf{1}_{1\times K}\Lambda\mathbf{U}^T = \mathbf{0}_{K\times 1} \tag{3.11}$$

$$\forall p^{(\ell)} \in (0, p^{(0)}] \quad \mathbf{1}_{1\times K}[\Lambda \bullet \mathbf{q}(p^{(\ell)})]\mathbf{1}_{d_v\times 1} < p^{(\ell)}. \tag{3.12}$$

Here, $p^{(0)}$ is the average message error rate of the first iteration, (3.12) ensures convergence by forcing $p^{(\ell+1)} < p^{(\ell)}$ and (3.10) ensures that Λ sums to one. In (3.11),

$\gamma = [\gamma_1, \dots, \gamma_K]$, and this constraint ensures that the fraction of bits transmitted on subchannel j is indeed γ_j.

The above formulation is not a standard MILP because in (3.12), $p^{(0)}$ depends on Λ. Notice that at the beginning of the first iteration, channel messages are propagated to the check nodes by the variable nodes. Therefore, the pdf of the messages at the input of the check nodes (and as a result $p^{(0)}$) is influenced by Λ. One strategy for solving this difficulty is to overlook this dependency; start with some approximation of $p^{(0)}$; formulate the problem as a MILP and after finding the optimum Λ update $p^{(0)}$. Now, we can solve the problem with the updated $p^{(0)}$ again. Our experiments show that this is an effective strategy and in many cases requires only a couple of rounds of MILP. Also, from (3.4), any change in Λ changes $(p^{(\ell)}, q^{(\ell)})$ pairs. That is, $\mathbf{q}(p^{(\ell)})$ is itself influenced by the design parameters $\lambda_{j,i}$. For conventional channels, it is shown in [36] that the direct effect of degree distribution on $q^{(\ell)}$ is much less than the effect through (3.4). Therefore, one can overlook the dependency of $\mathbf{q}(p^{(\ell)})$ on $\lambda_{j,i}$'s when $\lambda_{j,i}$'s undergo a small change.

Hence, the optimization can be performed as an iterative MILP. To this end, we need to ensure that the changes made in $\lambda_{j,i}$'s are small. This constraint can be imposed in various ways, but we are only interested in linear constraints. A simple way is to guarantee that in each round of optimization the amount of change in every $\lambda_{j,i}$ is smaller than a certain value ϵ. So, if the current value of the design parameters are $\overline{\Lambda} = [\overline{\lambda_{j,i}}]$, we have $(1-\epsilon)\overline{\Lambda} \leq \Lambda \leq (1+\epsilon)\overline{\Lambda}$ as an extra set of linear constraints on Λ.

3.3.2 Irregular Allotted Code

Semi-regularity of the codes was enforced to reduce the complexity of a search-based code design. We showed that instead of exhaustive search, design of semi-regular allotted codes can be done based on an iterative MILP. In the MILP formulation, however, semi-regularity of the codes obliged us to use integer parameters. Without the integer parameters, the problem could be solved as a linear program which is much more efficient. Also, the number of design parameters is reduced.

Interestingly, relaxing the constraints that were put forth to ensure semi-regularity will make the whole range of irregularity available to all subchannels. In other words, since semi-regular codes are special cases of irregular codes, this relaxation can only result in improved performance.

The code design formulation is similar to (3.5), but the optimization is performed on Λ, and constraints (3.6, 3.7, 3.8) are no longer required.

Numerical results, included in Section 3.5, support that in addition to the simpler design procedure, irregular LDPC codes perform better than semi-regular codes.

3.4 Absence of Channel Information at the transmitter

In the previous section, we discussed efficient design of allotted codes. We also mentioned that allotted codes require channel information at the transmitter. But in reality, many communication systems do not have this information.

In this section, we study two cases: (1) when CSI is absent at the transmitter, but available at the receiver and (2) when CSI is absent at both the transmitter and the receiver sides. In both cases, since CSI is not available to the transmitter, conventional codes are to be used.

One interesting property of many nonuniform channels is that the variation of the overall performance of the channel is much less than the variation of the subchannels. For example, suppose there are N_f frequency tones in an OFDM system and variance of the $\mathrm{SNR_{dB}}$ of every frequency tone is σ^2_{SNR}. The capacity of each frequency tone is

$$C_i = \log_2(1 + \mathrm{SNR}_i).$$

Using $C_i \approx \log_2(\mathrm{SNR}_i) = \mu\ \mathrm{SNR}_{\mathrm{dB}i}$, where $\mu = \frac{1}{10\log_{10} 2} = 0.3322$, we have

$$C_{\mathrm{avg}} \approx \mu \sum_i \frac{\mathrm{SNR}_{\mathrm{dB}i}}{N_f}$$

and therefore

$$\mathrm{Var}(C_{\mathrm{avg}}) = \frac{\mu^2\ \sigma^2_{\mathrm{SNR}}}{N_f^2}.$$

In practice N_f is a large number, thus it is a valid assumption that the capacity of every channel realization is almost equal to the average capacity. Therefore, even without CSI, the transmitter and the receiver can agree on a suitable code rate.

With no CSI, bits should be interleaved at the transmitter. By interleaving, we make sure that variable nodes of different degrees are transmitted through different subchannels, so that the average channel performance is observed. Without interleaving, chances are that nodes of different degrees are assigned to subchannels in an unfavorable way, resulting in poor performance.

It is known that the capacity of a parallel channel does not depend on the knowledge of CSI at the transmitter [37,38]. Thus, in the absence of CSI at the transmitter, with conventional LDPC codes one can approach the same capacity as allotted codes with CSI. One important question arises here. With both conventional and allotted codes approaching the same capacity, is there any benefit in using allotted codes? In the next section, by revisiting the fact that capacity does not change with availability of CSI at the transmitter, we discuss situations that allotted codes outperform conventional ones.

3.4.1 Only the receiver knows the channel

When the receiver knows the channel, it can recognize each one of the subchannels and calculate correct a posteriori probabilities. Since it is common practice to work with LLRs, it is more convenient here to state the capacity of each bit-channel as a function of its LLR distributions. Here, by capacity of each bit-channel (C_j), we mean the mutual information between the input and the output of the bit-channel when its input is equally likely zero or one. Thus we have

$$\begin{aligned} C_j &= H(X_j) - H(X_j|Y_j) \\ &= 1 - p(X_j = 0)E[\log_2 \frac{1}{p(X_j|Y_j)}|X_j = 0] - p(X_j = 1)E[\log_2 \frac{1}{p(X_j|Y_j)}|X_j = 1] \\ &= 1 - \frac{1}{2}E\left[\log_2\left(1 + \frac{p(X_j = 1|Y_j)}{p(X_j = 0|Y_j)}\right)|X_j = 0\right] - \frac{1}{2}E\left[\log_2\left(1 + \frac{p(X_j = 0|Y_j)}{p(X_j = 1|Y_j)}\right)|X_j = 1\right] \\ &= 1 - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1 + \mathrm{e}^{-m}) f_j^0(m)dm - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1 + \mathrm{e}^{m}) f_j^1(m)dm, \quad (3.13) \end{aligned}$$

where $f_j^0(\ell)$ and $f_j^1(\ell)$ denote the LLR pdf when the transmitted bit is zero and one, respectively[2]. It is evident that because the receiver can compute correct LLRs, the LLR pdfs of the interleaved channel will be

$$\begin{cases} f_{\text{mix}}^0(x) = \frac{1}{N}\sum_{j=1}^{N} f_j^0(x) \\ f_{\text{mix}}^1(x) = \frac{1}{N}\sum_{j=1}^{N} f_j^1(x) \end{cases} \quad (3.14)$$

[2]When binary modulation is used, because of the channel symmetry, (3.13) renders to the more familiar form given in [15], i.e., $C_j = 1 - \int_{-\infty}^{+\infty} \log_2(1 + \mathrm{e}^{-m}) f_j(m)dm$.

After interleaving at the transmitter, the average capacity is

$$C_{\text{avg}} = 1 - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+\mathrm{e}^{-m}) f_{\text{mix}}^{0}(m)dm - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+\mathrm{e}^{m}) f_{\text{mix}}^{1}(m)dm \tag{3.15}$$

$$= \frac{1}{N}\sum_{j=1}^{N} C_j. \tag{3.16}$$

which in agreement with literature [37] shows that the capacity does not change with the absence of CSI at the transmitter.

Conventional LDPC codes are shown to be able to perform very close to the capacity of many channel models [8, 19, 22]. Thus, an important question is whether or not allotted codes—which require channel information at the transmitter—have any superiority over conventional codes.

To answer this question, first notice that for (3.15) to hold, the receiver must be able to recognize subchannels so that f_{mix} in (3.15) becomes the average of original f_j's according to (3.14). Therefore, the result which says interleaving at the transmitter does not change the capacity is valid only when the nonuniformity of the channel is fully taken into account by the receiver (i.e., when an optimal decoder is employed). As a radical example of suboptimal decoding, consider the case of hard decoders. For conventional codes, due to hard decoding, different quality of subchannels at the decoder is completely ignored, as if CSI did not exist at either side. Absence of CSI at both the transmitter and receiver results in an unavoidable capacity loss on top of performance loss due to suboptimal decoding. However, for an allotted code—even under hard decoding—the existence of channel knowledge at the transmitter prevents the above mentioned capacity loss. The difference lies in the ability of allotted codes to utilize the reliability difference of subchannels in the code structure.

The difference between the performance of allotted and conventional LDPC codes should decrease as the decoding precision increases, but will not vanish as long as the decoding is suboptimal. Conventional LDPC codes can be designed to provide close to capacity performance under sum-product decoding for many channels [8, 22, 31]. Therefore, for such cases, sum-product decoding has an almost optimal performance. As a result, we do not expect a meaningful difference between the performance of allotted and conventional LDPC codes under sum-product decoding for highly optimized codes. Thus, under suboptimal decoding or tight constraints on the maximum node degree in the code (which prevent close-to-capacity performance) we expect allotted codes to provide a better performance than conventional ones. Notice that conventional codes are special cases of allotted codes. This discussion is supported by the examples in Section 3.5.

3.4.2 Neither side knows the channel

Consider a nonuniform channel when the instantaneous parameters of its subchannels are not known either to the transmitter or to the receiver. Similar to the previous case, interleaving is the natural choice. After interleaving the bits, the nonuniform channel can be modeled as a uniform one.

Suppose the original channel is made up of K subchannels where subchannel j carries γ_j fraction of the bits and characterized by the conditional probability $p_j(Y|X)$. Also assume that all subchannels share the same input alphabet. It is known that when the receiver does not know CSI, the capacity of the system is reduced. Unlike the preceding case, the receiver is only aware of the average behavior of the channel which means that the receiver sees the channel as a uniform one with the following conditional probability,

$$p(Y|X) = \sum_j p(\text{subchannel } j)p(Y|X, \text{subchannel } j) = \sum_j \gamma_j p_j(Y|X). \tag{3.17}$$

Since capacity is a concave function of conditional probabilities of the channel, we have

$$C\big(p(Y|X)\big) = C\left(\sum_i \gamma_i p_i(Y|X)\right) \leq \sum_i \gamma_i C\big(p_i(Y|X)\big),$$

which shows an inevitable capacity loss in the interleaved channel. Nonetheless, our example shows that one can approach the reduced capacity using conventional LDPC coding.

3.5 Numerical Results

3.5.1 Semi-regular and irregular allotted codes

As the first example, we revisit a code design problem solved in [25], i.e., a volume holographic memory systems. The system can be considered as a set of parallel channels having different noise powers. It is assumed that there are four parallel channels ($K = 4$). The noise is assumed to be Gaussian. Hence, the system can be modeled as a set of four binary-input additive white Gaussian noise channels. The relative SNRs of different regions are SNR2 - SNR1 =1.61 dB, SNR3 - SNR1 =2.80 dB, SNR4 - SNR1 =3.74 dB.

We want to design a code of rate 0.85. Similar to [25], in order to prevent the error floor (because a bit error rate (BER) of at least 10^{-12} is needed), we avoid degree-two variable nodes in the graph.

Solving the suggested MILP, the same semi-regular code obtained in [25] is found as follows:

$$\begin{aligned} \rho_{40} &= 1, \\ \Lambda &= \begin{bmatrix} 0 & 0 & 0.1250 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1667 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2917 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4167 \end{bmatrix}. \end{aligned}$$

Using density evolution, we obtain the threshold for this code as $\frac{E_b}{N_0} = 2.73$dB. We design an irregular allotted LDPC code with the same threshold using the iterative LP method. The following code is obtained:

$$\begin{aligned} \rho_{26} &= 1, \\ \Lambda &= \begin{bmatrix} 0 & 0 & 0.1407 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2300 \\ 0 & 0 & 0.2097 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2097 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2097 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

The rate of the irregular code is $R = 0.857$, which is slightly improved, but more importantly, the design procedure was done considerably faster. For a code length of 10^5 the bit error rate curves are obtained for these two codes (Fig. 3.3), which show that the irregular code has a better performance by a margin of 0.13 dB [3].

[3] It deserves mention that [25] suggests a special method of making the Tanner-graph. In our experiments, however, both codes are randomly constructed and cycles of length up to four are removed.

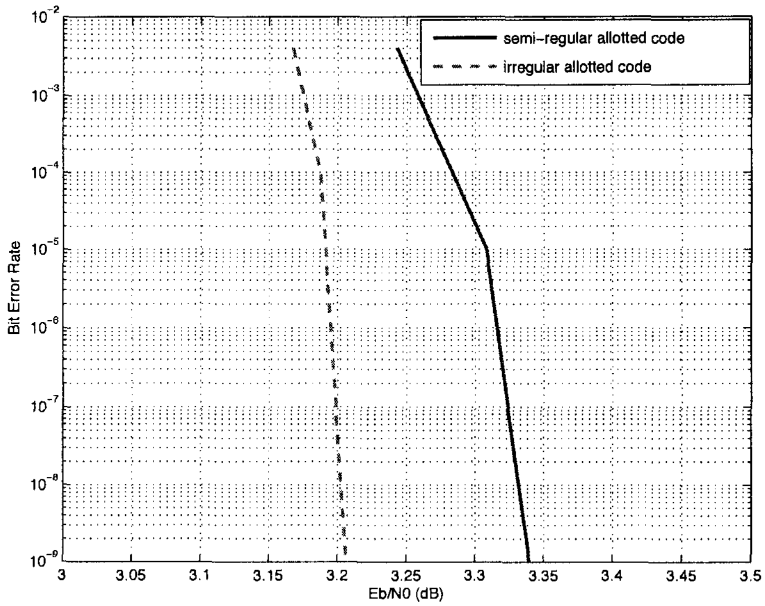


Figure 3.3: Bit error rate curves for the irregular allotted LDPC code and semi-regular allotted LDPC codes designed in Section 3.5.1. Both codes have a length of 10^5. The curves are obtained through Monte Carlo simulation.

For a given rate, the number of edges of the Tanner-graph of a given code is considered a measure of decoding complexity [39, 40]. In fact, the number of message updates per iteration grows almost linearly with the number of edges. The number of edges normalized per information bit for the irregular code is 4.17 whereas this number for the semi-regular code is 7.06. In other words, the irregular code's decoding complexity is also reduced by more than 40%.

3.5.2 Code design for an OFDM system

In the following example we study coding solutions under different scenarios regarding the availability of CSI for an OFDM system. The average capacity of the actual and interleaved channels are shown by C_{avg} and C_{int}, respectively.

The basic feature in an OFDM system is that it transforms a frequency selective channel into a group of independent (orthogonal) parallel narrowband AWGN channels [33]. Because of the frequency selectivity of the channel, some frequency tones, or, in the context of nonuniform channels, some subchannels, experience significant attenuation while other frequency tones provide much better SNR. This extensive nonuniformity has been considered the main difficulty in the code design.

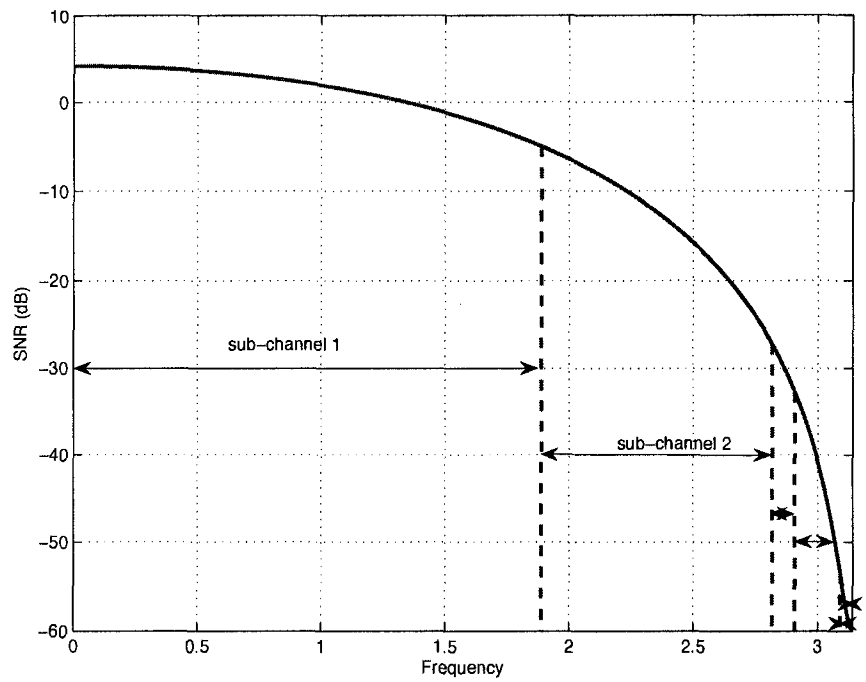


Figure 3.4: Proakis B channel with the frequency tones grouped together in six subchannels as in [1]

Due to the nonuniformity of OFDM channels, allotted LDPC coding has been used for OFDM channels [1, 26, 27]. Here, motivated by our results, we design and compare LDPC codes for different scenarios.

Consider designing an LDPC code with code length **L** and rate R, for an OFDM system with N_f sub-carriers. Assume the OFDM system has proper cyclic insertion and sampling. Coded bits are modulated by quadrature amplitude modulation (QAM) with fixed constellation size q for all frequency tones. Also, bit-interleaved coded-modulation [34] is used.

To comply with the experiments in [1], we use Proakis B as the channel and a 4-QAM constellation. Fig. 3.4 shows the frequency response of the channel. We design five LDPC codes for this channel: two irregular allotted LDPC codes with different maximum degree, similarly, two conventional LDPC code assuming CSI at the receiver, and one conventional LDPC code assuming the absence of CSI at the receiver. Using sum-product algorithm with 11-bit precision (2048 fixed LLR levels), each code should converge to a message error rate of 10^{-6} in less than 400 iterations.

CSI available to both sides

The channel is known to both the transmitter and the receiver. In order to reduce the number of subchannels, we can group frequency tones into five groups according to their SNRs, such that, for example, every group contains one sixth of frequency tones. In order to have a better comparison with [1], we use a similar grouping: $\gamma = [0.6013,\ 0.2936,\ 0.0281,\ 0.0563,\ 0.0083,\ 0.0124]$. Also, since the threshold of the code designed in [1] is $E_b/N_0 = 3.03\text{dB}$[4], we design all of the following codes for the channel with SNR = 3.03dB. With 4-QAM signalling, the capacity of this channel is $C_{\text{avg}} = 0.5129$.

Allowing a maximum degree of 20 in the code, and using a regular check degree of 7 ($\rho_7 = 1$), following the proposed algorithm in Section 3.3, the result of optimization is the following code: $R = 0.5069$, $\lambda_{1,2} = 0.3241$, $\lambda_{1,20} = 0.2428$, $\lambda_{2,2} = 0.0868$, $\lambda_{2,3} = 0.1249$, $\lambda_{3,3} = 0.0244$, $\lambda_{4,3} = 0.0086$, $\lambda_{4,6} = 0.0499$, $\lambda_{4,20} = 0.1024$, $\lambda_{5,6} = 0.0144$, $\lambda_{6,6} = 0.0216$, which achieves 98.8% of the capacity. To see how a tighter constraint on complexity affects the performance, the coding problem is solved again under the constraint that the maximum allowed node degree is 6. The result is $R = 0.487$: $\rho_6 = 1$, $\lambda_{1,2} = 0.1005$, $\lambda_{1,3} = 0.3744$, $\lambda_{1,6} = 0.1225$, $\lambda_{2,2} = 0.1875$, $\lambda_{2,6} = 0.0101$, $\lambda_{3,6} = 0.0548$, $\lambda_{4,6} = 0.1098$, $\lambda_{5,6} = 0.0162$, $\lambda_{6,6} = 0.0242$, which achieves 95.1% of the capacity.

As Table 3.1 shows, there is not a considerable gain in terms of code threshold compared to the code optimized in [1]. It has to be mentioned that the method used in [1] assumes the liberty of having a very large number of frequency tones. This is a fair assumption in the case of OFDM systems, which means γ is not forced by the system. Our method, however, is aimed at more general cases.

CSI available only to the receiver

Now, suppose the transmitter does not know the quality of frequency tones, but the receiver has a good estimate of the SNR in each frequency tone. Hence, we can only use a conventional code, and, as discussed before, we do not expect a significant degradation. Again, regular check nodes with degree 7, and a maximum variable

[4] E_b/N_0 is, simply put, the energy of the signal per information bit divided by the energy of the noise

Table 3.1: Code rates obtained for various LDPC codes.

LDPC code	[1]	irregular allotted	irregular allotted	conventional	conventional
Maximum degree	20	20	6	20	6
R	0.500	0.507	0.487	0.491	0.461
$(E_b/N_0)^*$ (dB)	3.03	2.97	3.14	3.11	3.38
Gap to capacity	2.52%	1.15%	4.90%	4.3%	10%

node degree of 20 are used. The result is the following degree distribution: $\lambda_2 = 0.3007$, $\lambda_3 = 0.2517$, $\lambda_6 = 0.1877$, $\lambda_8 = 0.0006$, $\lambda_{12} = 0.0310$, $\lambda_{13} = 0.0341$, $\lambda_{20} = 0.1932$, whose rate is $R = 0.4910$ which is 95.7% of the capacity. This is 96.9% of the allotted code rate. This supports the idea that channel information is not necessary at the transmitter side, when the decoding algorithm performs close to optimal.

This time, a tighter set of constraints can dramatically affect the performance. Again, under the constraint that the maximum allowed node degree is 6, we obtain the following code: $\rho_8 = 1$, $\lambda_2 = 0.3544$, $\lambda_3 = 0.1479$, $\lambda_6 = 0.4977$, whose rate is $R = 0.4614$ or 90.0% of the capacity.

So it is clear that if the allowed complexity, thus the allowed maximum degree, of the code is under a tighter constraint, the difference between allotted and conventional LDPC codes becomes more pronounced.

CSI not available

Finally, suppose the receiver does not distinguish between subchannels, but knows the conditional probabilities of the mixed channel (3.17). Calculating LLRs from conditional probabilities is a straightforward task

In this example, we obtain $C_{\text{int}} = 0.4172$ which is 81.3% of the actual capacity ($C_{\text{avg}} = 0.5129$) based on correct LLR computation. This is interesting considering the vast difference among the quality of frequency tones. With $d_c = 6$ and $d_v = 10$ and other conditions similar to the preceding cases, the optimum code is $R = 0.3974$: $\rho_5 = 0.2$, $\rho_6 = 0.8$, $\lambda_2 = 0.3039$, $\lambda_3 = 0.2670$, $\lambda_4 = 0.0251$, $\lambda_{10} = 0.4040$. This code achieves 95.3% of the reduced capacity or 77.4% of the actual capacity of the channel (C_{avg}).

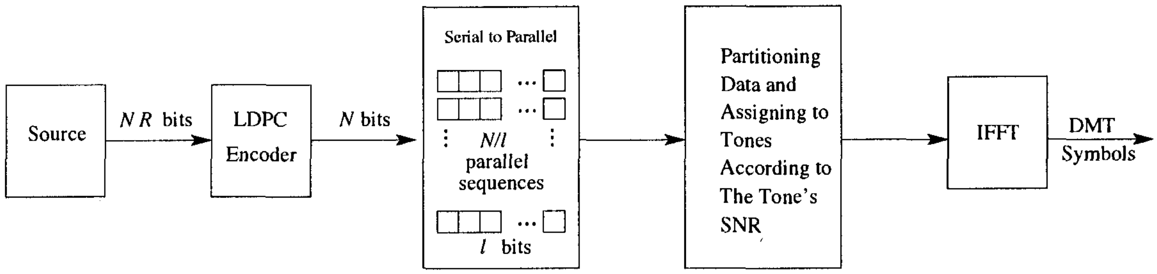


Figure 3.5: Block diagram of the transmitter. The partitioning is done based on the existing knowledge about the DMT channel.

We see that although mixing subchannels is a suboptimal solution, considering the reduced complexity of the system and that an LDPC code can approach this reduced capacity, it might be a viable option.

3.6 System Design for Power-Line Channels

In this section, first we describe the overall structure of a typical DMT over power-line, and then perform LDPC code optimization. We also design a conventional LDPC code to show the improvement obtained by optimized allotted LDPC codes.

The overall structure of the coding system is shown in Fig. 3.5. The LDPC encoder takes $R \cdot N$ bits from the source and produces N coded bits regardless of the nonuniformity of the channel. Here, R is the code rate. The N coded bits are then broken to N/l sequences. Each sequence has l bits and represents one of the 2^l points of a QAM constellation. These N/l sequences are assigned to equivalent bit-channels according to Λ.

Once all the binary sequences for all tones are ready, each binary sequence is mapped to a complex symbol (according to the labeling scheme) and using inverse Fast Fourier transform (IFFT) a DMT symbol is created. Notice that one LDPC codeword may consist of multiple DMT symbols. Since the number of bits in a DMT symbol depends on the channel realization, the length of the LDPC may not be an integer multiple of DMT symbols. However, the code block-length N is usually much larger than the number of bits in one DMT symbol. Therefore, one can fill the LDPC codeword with as many as possible DMT symbols and fill the remainder of the codeword with zeros.

At the receiver, this process is reversed. When the LLR value for every bit of the codeword is computed the decoding process starts. Similar to the encoding, the decoding is also independent of the nonuniform channel. Hence no modification on the decoder and the encoder of the code is required.

There are many iterative decoding algorithms available for LDPC codes. Although in the next section, we optimize the LDPC code under sum-product decoding, our methodology based on the proposed recursive LP is quite general and can be applied to any decoding algorithm for which a density-evolution analysis [19] is possible.

We use the channel model proposed in [41] for power-line communication and 64-QAM constellation for all tones. The distribution of SNR in different tones is shown in Fig. 3.6. Note that the effects of impulse noise are neglected here. While the channel model of [41] considers water-filling, it has to be mentioned that water-filling does not affect our approach as the coding solution is based on the channel SNR distribution.

In order to avoid bit-channels with a very low capacity, tones that have an SNR less than a threshold should carry no information. For 64-QAM signalling, we may decide not to use any tone with an SNR less than 1.2 dB. This way, there will be no bit-channel with a capacity less than 0.2 bits/symbol.

This approach is employed to reduce the overall complexity and has minor effect on the overall performance of the system, because low-capacity bit-channels have minor effect on the overall capacity of the system. For example, assume that we have 1000 bit-channels. Assume that there are 200 bit-channels with capacity less than 0.2. Instead of having 1000 bit-channels with average capacity of 0.42, we would rather have 800 bit-channels with average capacity of, say, 0.5 and neglect the 200 bit-channels with worst capacity. This leads to a considerable reduction of complexity with a slight degradation. It should be emphasized that even without this consideration, the proposed coding solution works perfectly.

Since the number of active tones is relatively large, if K (the number of parallel subchannels) is relatively small, one can expect an average behavior in almost all channel realizations. Therefore, the coding solution will be robust to changes in the channel as long as $K \ll$ (number of active frequency tones).

We use $K = 4$ and the capacity range for subchannels are selected to be [0.2, 04), [0.4 0.6), [0.6 0.8) and [0.8, 1). On a 64-QAM signalling, these capacity ranges map to the following SNR ranges respectively: [1.2dB 6.5dB), [6.5dB 10.8dB), [10.8dB 14.8dB) and [14.8dB $+\infty$). From the distribution of the SNR (Fig. 3.6) it can be easily found that $\gamma_1 = 0.3364$, $\gamma_2 = 0.2949$, $\gamma_3 = 0.2022$, and $\gamma_4 = 0.1665$.

When the channel condition, the constellation size and the labeling scheme are known, the density of LLR messages of the channel can be found via Monte Carlo simulation. This provides an accurate analysis for a DMT system whose frequency tones are distributed according to the typical distribution depicted in Fig. 3.6. Then LLR distribution is used in density evolution.

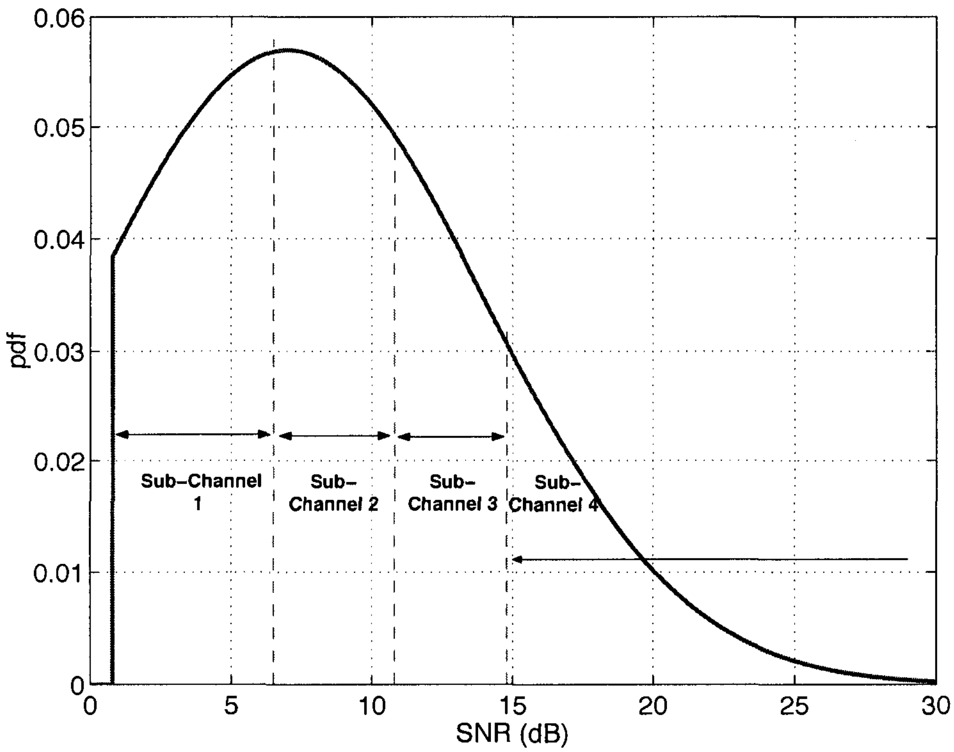


Figure 3.6: Distribution of SNR in frequency tones of the power-line channel. Frequency tones are grouped into four subchannels according to their SNR.

All of the codes in this section are designed so that they converge to target error rate of 10^{-7} in less than 400 iterations of sum-product decoding with 11-bit precision. The effect of 11-bit decoding and 400 iterations is in implementation of "discrete density evolution" [22]. Choice of 400 iterations and 11-bit decoding is arbitrary and the approaches of this work are readily applicable to other numbers if needed.

Allowing a maximum node degree of 10 in the code, the optimized degree distribution for this channel is $\rho = \{\rho_8 = 1\}$ and $\Lambda = \{\lambda_{1,2} = 0.0058, \lambda_{1,3} = 0.1794, \lambda_{1,10} = 0.2003, \lambda_{2,2} = 0.1449, \lambda_{3,2} = 0.0994, \lambda_{4,2} = 0.0099, \lambda_{4,10} = 0.3603\}$. This code has a rate of $R = 0.4916$. This means that more than 96% of the capacity of the channel $C = 0.5077$ bits/symbol is achieved with a code whose maximum node degree is 10.

The above code was designed on a simplified channel because of the $K = 4$ assumption. In order to have a sound comparison with conventional codes the code is tested on the actual channel with successful convergence.

To see how an optimized allotted LDPC code can outperform a conventional LDPC code, we do similar optimization for a conventional LDPC code. The result is, $\rho = \{\rho_8 = 1\}$ and $\lambda = \{\lambda_2 = 0.2564, \lambda_3 = 0.0443, \lambda_{10} = 0.6993\}$. This code has a rate of $R = 0.4129$ which is no more than 81.3% of the capacity of the channel. Notice that in this case the nonuniformity of the channel is still used. In fact, the channel state information is used at the receiver to calculate correct LLRs. That is to say, the receiver recognizes different frequency tones and knows their correct SNRs. So, the only difference is that all subchannels are forced to use the same degree distribution.

This comparison shows that for a practical maximum degree of 10, the conventional LDPC code performs well below allotted LDPC codes. It is interesting that the improvement is obtained at almost no extra cost. Nevertheless, it should be pointed out that the difference becomes less significant if one allows impractical degree distributions.

Repeating the optimization with a maximum variable node degree of 25, we obtained a rate 0.4707 code with $\rho = \{\rho_8 = 1\}$, $\lambda = \{\lambda_2 = 0.2612, \lambda_3 = 0.1971, \lambda_5 = 0.0244, \lambda_6 = 0.1057, \lambda_{12} = 0.0204, \lambda_{25} = 0.3912\}$. This rate is closer to capacity but still less than the rate of an allotted code with much less complexity.

Considering that channel state information is available at the transmitter and the receiver in power-line DMT channels, allotted LDPC codes seem to be the natural choice.

3.7 Conclusion

It was shown that semi-regular allotted LDPC codes can be designed via mixed integer linear programming. The proposed approach can be simplified to linear programming when the semi-regularity constraint is removed.

Since allotted codes require the channel state information at the transmitter and channel knowledge at the transmitter does not change the capacity, a natural question is whether allotted codes provide any benefit over conventional LDPC codes. We argued that under optimal decoding the performance gap between allotted and conventional codes is minor. This gap, however, can be quite significant if suboptimal decoding is used or when the maximum node degree allowed in the code is small. Therefore, when channel state information is available at both the transmitter and receiver sides, it is better to use allotted LDPC codes, since they approach the capacity with smaller node degrees in the code and also because they outperform conventional codes with suboptimal decoders. We also investigated the case when neither the transmitter nor the receiver has the channel knowledge. While a capacity loss is inevitable, the reduced capacity appears to be approachable by conventional LDPC codes.

Chapter 4

Universal LDPC Codes

Design of codes for specific channels is a well studied subject. A code optimized for one channel, however, may not be suitable if used on another channel [42, 43]. Therefore, more recently, there has been an emerging interest in universal codes, i.e., codes that provide good performance for a multitude of channels [44–47]. Such codes reduce system complexity by removing a need for the CSI at the transmitter and frequent code changes in the communication system. They also allow for once-and-for-all coding solutions[1].

4.1 Introduction

The existence of universal error correcting codes can easily be deduced from the Shannon's channel coding theorem. Consider all random codebooks of $(2^{nR}, n)$, i.e., all codes of length n and rate R, indexed from 1 to $2^{n(1-R)2^{nR}}$ and define $n_u \triangleq n2^{n(1-R)2^{nR}}$. A universal $(2^{n_u R}, n_u)$ code can be constructed as follows. To encode $n_u R$ bits, parse the sequence to n_u/n subsequences of length nR. Then, encode the ith subsequence with the codebook number i. According to [3, Theorem 8.6.1], the average bit error rate of a jointly-typical decoder over all $(2^{nR}, n)$ codebooks on any given channel with $C > R$ approaches zero as $n \to \infty$. Thus the newly constructed $(2^{n_u R}, n_u)$ code is a capacity achieving universal code.

Therefore, with a random code and a jointly-typical decoder (or an optimal maximum likelihood decoder), where the decoder is adjusted for each channel, random codes with universal property can be constructed. However, the complexity

[1] A version of this chapter has been submitted for publication [48]

of a jointly-typical or a maximum likelihood decoder grows exponentially with the length of the codeword (n). Thus, such universal coding solutions are not practical.

Low-density parity-check (LDPC) codes [5] are known to be extremely strong error correcting codes. In fact, if carefully designed, these codes can approach the capacity of most channel types [8, 22] with practical complexity. The question is whether LDPC codes under iterative decoding have good universal properties and whether LDPC codes with stronger universal properties under iterative decoding can be constructed. Interestingly, various authors have observed good "universal properties" of these codes [12–14, 46, 47].

Chung [12] notices that LDPC codes designed for the Gaussian channel have good performance on some other channels such as the Rayleigh channel. Jin *et al.* [46] define a universal LDPC code as a code with similar performance over a class of channels, in which the performance is measured by the threshold of the code in terms of mutual information (in bits/sec/Hz). They observe that LDPC codes have good universal properties on fading channels with variant fading rates. In a more general setup, Shi and Wesel [47] study the universal properties of finite block length codes.

Peng *et al.* [14] study design of LDPC codes that are suitable for a number of channels (in this case, the Gaussian channel, the BEC and the Rayleigh channel). For a given set of channels, they find that usually one channel can be chosen as the surrogate for the set. They propose using a code which is designed for the surrogate channel. This approach results in codes which work satisfactorily on all the given channels, but not necessarily on other channel types.

Despite some observations on universality of LDPC codes on various channels, the performance of a code designed for one channel, can be prohibitively worse on another channel with similar capacity. For example, an LDPC code with maximum node degree of 100 that achieves more than 99.7% of the capacity of a BEC with capacity $C = 0.5$ taken from [49], does not converge to an acceptable error rate over a BSC with capacity $C = 0.63$. That is, even 80% of the capacity of the BSC is not achieved by this code.

In this work, we study design of LDPC codes that have strong universal properties. That is, for a given channel capacity C, we seek LDPC codes that provide good performance on all channels which exhibit this capacity. To this end, we take two steps that form the main body of this chapter: first, code design for convex combination of N channels of equal capacity, and second, decomposition of channels into a number of subchannels with similar capacity.

In Section 4.3, we investigate the code design problem for convex combinations of two equal-capacity channels, namely CH_1 and CH_2. By convex combination of two channels, we mean a scenario where γ fraction of transmitted bits pass through CH_1 and the other $1-\gamma$ fraction pass through CH_2, where $\gamma \in [0,1]$. For a specific γ, this linear combination specifies a channel, CH_γ, for which one can easily design a good code. The problem that we try to solve in this Section is quite different. Our goal is to find a code which works for all values of γ. Such a code is universal on all convex combinations of CH_1 and CH_2.

We show that under a mild assumption, to design a code for all convex combinations, it is sufficient to guarantee the convergence of the code only on its constituent channels (CH_1 and CH_2). The same result would apply to any number of equal-capacity subchannels. That is, a code designed to converge on N channels, would converge on the convex hull of these N channels.

A consequence of this result is that if a set of basis channels of capacity C can be found, a code designed only for the basis channels is universal. This is because any channel with capacity C can be expressed as a convex combination of these basis channels.

In Section 4.4, a channel decomposition method is suggested. Our method decomposes a BSC with capacity C into a number of basis channels of the same capacity with nonnegative coefficients. Because of the identical capacity of the basis channels, this decomposition method is not similar to different existing channel decomposition techniques—most famous, perhaps, is channel decomposition over BSCs with various capacities [16, 50]. We show that our technique is exhaustive, i.e., all channels of a given capacity can be spanned with nonnegative coefficients over the suggested basis.

Therefore, for any given capacity C, a code designed only for the basis channels that are obtained from our channel decomposition method is expected to have strong universal behavior over all channels with capacity C. Code design for the basis channels and simulation results are provided in Section 4.5. The yardstick that we use to measure the success of our codes is the achievable percentage of the channel capacity. We propose an upper bound on the achievable rates by universal LDPC codes and we show that our codes approach this upper bound. Also, through simulation results, we show that our codes have significantly better universal properties compared to codes with similar rate designed for specific channels.

Finally, it is important to emphasize that throughout this chapter, we assume that all channels are memoryless BISO and fed with equiprobable binary inputs.

4.2 A Short Note on LDPC Code Design

An ensemble of irregular LDPC codes is usually represented by variable and check degree distributions, $\lambda(x) = \sum \lambda_i x^{i-1}$ and $\rho(x) = \sum \rho_i x^{i-1}$, respectively.

LDPC code design on a given channel means optimizing the check and variable degree distributions of the code according to a cost function. Various cost functions are considered in the literature [8, 51]. The most common two approaches are threshold maximization and rate maximization.

In threshold maximization, the rate of the code is fixed and for a given channel type, the goal is to find the code which works under the worst channel quality. This approach minimizes the gap from the channel capacity for a fixed code rate. In rate maximization (similar to the approach in Chapter3), the channel parameters are fixed, and the goal is to obtain the code with maximum rate that provides reliable communication on the given channel. This approach minimizes the gap from the capacity for a fixed channel.

The focus of this chapter is on the codes that are suitable for a multitude of channels rather than a single one. The rate-maximization approach fits our purpose well, because it allows for a natural generalization from a single channel to a finite number of them. In other words, two or more channels with identical capacity are given and the objective is to find the code with maximum rate that performs well on all of these channels. Using this approach, code design is very similar to what

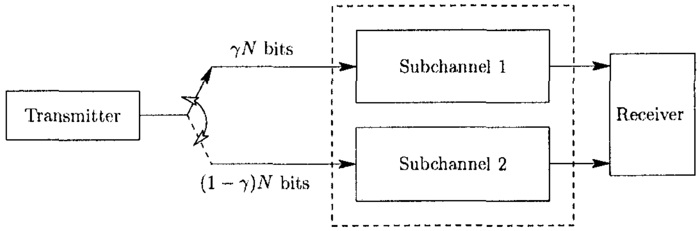


Figure 4.1: Convex combination of two subchannels, where γ is the fraction of bits that go through subchannel 1.

has been discussed in the literature in various forms [52, 53].

In this chapter density evolution is used for the purpose of code design [8] which was introduced in Chapter 2. Though to some extent repetitious, a brief review of the method with a little embellishment to allow for code design for two channels is presented in Section 4.3.2.

4.3 Universal Codes for Convex Combinations of Two Channels

In this section, we consider codes that are universal for any convex combination of two channels. We start with the definition of convex combination of two channels.

4.3.1 Convex Combination of Two BISO Channels

Fig. 4.1 illustrates a communication channel composed of two subchannels, where γ fraction of bits pass through subchannel 1 and $1-\gamma$ fraction of bits pass through subchannel 2. We refer to this communication channel as a convex combination of subchannels 1 and 2.

Using LLRPDFs of subchannels, this combination of channels translates into linear combination of the corresponding LLRPDFs. That is,

$$f^{\gamma}(m) = \gamma f^{1}(m) + (1-\gamma) f^{2}(m), \tag{4.1}$$

where $f^{\gamma}(m)$, $f^{1}(m)$, and $f^{2}(m)$ denote LLRPDFs of the mixed channel, subchannel 1 and subchannel 2, respectively. For (4.1) to hold, the receiver must have CSI and compute correct LLRs (an assumption made throughout this chapter).

Similarly, based on PEPDF representation we have

$$g^{\gamma}(p) = \gamma g^{1}(p) + (1-\gamma) g^{2}(p), \tag{4.2}$$

where $g^{\gamma}(p)$, $g^1(p)$, and $g^2(p)$ denote PEPDFs of the mixed channel, subchannel 1 and subchannel 2, respectively.

One of the important properties of the linear combination of two LLRPDFs is that the same linear combination applies to the mutual informations, i.e.,

$$\mathcal{I}\left(f^{\gamma}(m)\right) = \gamma \mathcal{I}\left(f^1(m)\right) + (1-\gamma)\mathcal{I}\left(f^2(m)\right). \tag{4.3}$$

This equation holds, because according to (2.2), $\mathcal{I}\left(f^{\gamma}(m)\right)$ is a linear function of $f^{\gamma}(m)$ which itself is a linear combination of $f^1(m)$ and $f^2(m)$.

If γ is fixed, the linear combination defines a unique channel, for which one can design a good LDPC code. If γ is unknown, however, finding a code which works for all values of $\gamma \in [0, 1]$ is a challenging problem.

It is noteworthy that if the transmitter knows through which of the subchannels the next transmitted bit will pass, it can use two different channel codes, one for subchannel 1 and another one for subchannel 2 and fill the buffers on the fly. Thus, universal codes are not required. In most practical situations, however, the transmitter does not have such side information.

When both subchannels have capacity C, from (4.3) it is evident that the capacity of the convex combination is also C regardless of γ. Our goal is to design a single LDPC code which works for all values of γ (see Fig. 4.1). As we will see, solving this problem paves the way for finding universal LDPC codes that work for any channel with a given capacity C.

As a first step, let us assume that there are only two possible cases, $\gamma = 0$ and $\gamma = 1$. That is, we assume the channel is equivalent to one of its subchannels. It will be shown later that under a mild assumption, the code designed based on this assumption is suitable when γ can take other values as well.

4.3.2 Code design for two channels

When there are only two possible cases of $\gamma = 0$ and $\gamma = 1$, it is sufficient to find a code which works on both subchannels 1 and 2. Taking the rate maximization approach for code design and assuming that $\rho(x)$ is given, in order to maximize the code rate

$$R = 1 - \frac{\sum \rho_i / i}{\sum \lambda_i / i},$$

one has to maximize $\sum \lambda_i/i$, which is a linear objective function in terms of the design parameters $\{\lambda_i\}$. This optimization is subject to convergence of the code on these two subchannels as well as $\lambda(x)$ describing a valid code.

For code design, we use density evolution. Although, in this section we consider only two subchannels, we will consider a larger number of subchannels for the purpose of code design in the sequel. Thus, we formulate the design problem as a linear programming to allow for fast code design.

Using density evolution for each subchannel, one can obtain asymptotic behavior of the code over each subchannel. At each iteration of the density evolution, we have the density of LLR input messages (input to check nodes, output of variable nodes). We denote this density at iteration ℓ with $f_\ell(x)$. This density is the same for all edges, as the random structure of the LDPC code interleaves the messages. Assuming that the all-zero codeword is transmitted, this density defines a unique mutual information which is

$$I_{\text{in},\ell} = \mathcal{I}\left(f_\ell(m)\right). \tag{4.4}$$

Here, $I_{\text{in},\ell}$ represents the amount of mutual information at the beginning of the ℓth iteration.

Running density evolution, one obtains the density of LLR messages at the output of degree i variable nodes, $f_{i,\ell+1}(x)$, for all i. The input density to the $\ell+1$th iteration is therefore

$$f_{\ell+1}(m) = \sum_i \lambda_i f_{i,\ell+1}(m).$$

Thus, the input mutual information to the iteration $\ell + 1$ is [53, 54]

$$I_{\text{in},\ell+1} = I_{\text{out},\ell} = \sum_i \lambda_i \mathcal{I}\big(f_{i,\ell+1}(m)\big).$$

By removing the iteration index for simplifying the notations, the above equation can be rewritten as

$$I_{\text{out}} = \sum_i \lambda_i I_{\text{out},i},$$

where $I_{\text{out},i}$ denotes the output mutual information at degree i variable nodes.

Since, $f_{i,\ell+1}(m)$s are functions of $f_\ell(m)$ and $f_\ell(m)$ corresponds to $I_{\text{in},\ell}$ (or for simplicity I_{in}) according to (4.4), one can think of the mutual information at the output of degree i variable nodes, $I_{\text{out},i}$, as a function of I_{in}. We denote this function by $I_{\text{out},i}(I_{\text{in}})$. After running density evolution for M iterations, this function is known at M values of I_{in} visited by density evolution. Hence, one can use these M values of I_{in} to ensure the designed code exhibits satisfactory convergence behavior. Note that this gives rise to a linear program as follows,

$$\text{maximize} \quad \sum_{i=2}^{\max(d_v)} \frac{\lambda_i}{i}$$

$$\text{s.t.} \quad \lambda_i \geq 0 \quad \forall i$$

$$\sum_{i=2}^{\max(d_v)} \lambda_i = 1$$

Convergence on channel 1:

$$\sum_{i=2}^{\max(d_v)} \lambda_i I^1_{\text{out},i}(x) > x, \quad \forall x \in [C, 1)$$

Convergence on channel 2:

$$\sum_{i=2}^{\max(d_v)} \lambda_i I^2_{\text{out},i}(x) > x, \quad \forall x \in [C, 1),$$

where C is the mutual information at the beginning of density evolution which is equal to the capacity of both of the channels. Since we want to ensure convergence of the code only on two channels, there are two sets of constraints on the convergence behavior.

Example 1 *We seek a maximum-rate LDPC code which converges on both BEC and BSC with capacity* $C = 0.5$. *The code should have a fixed regular check node degree of 9, and a maximum variable node degree of 50. The code is supposed to converge to a target mutual information of* $1 - 2 \times 10^{-6}$ *in less than* 400 *iterations.*
Solution: $\lambda_2 = 0.1979$, $\lambda_3 = 0.1853$, $\lambda_5 = 0.0417$, $\lambda_7 = 0.0552$, $\lambda_8 = 0.2133$, $\lambda_{50} = 0.3066$, *Rate*=0.4702.

Notice that if the check node degree distribution is not given, the formulation is not different, except for the objective function which is not linear. This difficulty exists in the case of code design for a single channel too. A short discussion on how to choose $\rho(x)$ is presented in Section 4.5.1.

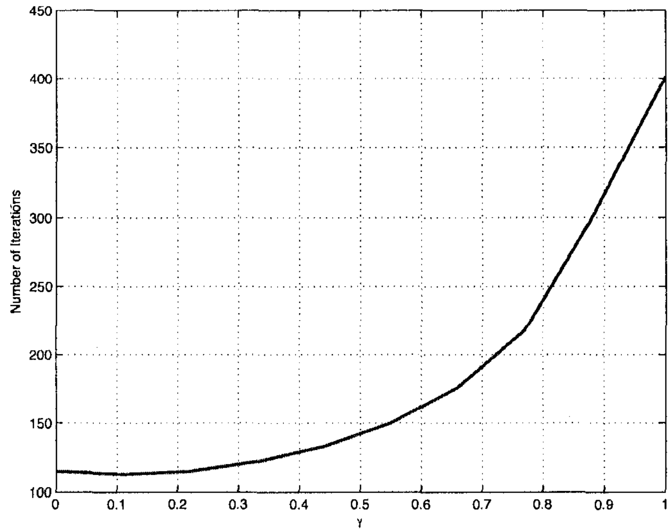


Figure 4.2: The number of iterations needed for the code of Example 1 to converge on different convex combinations of the BEC and BSC. Here, γ is the fraction of bits that pass through the BSC. Thus $\gamma = 0$ and $\gamma = 1$ represent the BEC and BSC respectively.

As a final remark, notice that if we needed convergence on N channels, we would have a similar formulation with

$$\sum_{i=2}^{\max(d_v)} \lambda_i I^j_{\text{out},i}(I_{\text{in}}) > I_{\text{in}}, \quad \forall I_{\text{in}} \in [C, 1),$$

repeated for $1 \leq j \leq N$, i.e., all N channels.

4.3.3 All convex combinations of two subchannels

A code for the special case that γ is either zero or one was designed in the previous section. In general, however, γ can take any value between zero and one. The number of constraints in the previous optimization problem grows almost linearly with the number of different channels (i.e., different possible values of γ) that the code should be guaranteed to work on.

Fortunately, our experiments show that if we guarantee convergence for two values of γ, the code will converge for all the values of γ in between. For different values of γ, Fig. 4.2 shows the number of iterations that the code designed in Example 1 needs to converge to the specified target mutual information. Not only does the code converge for every $\gamma \in [0, 1]$, but also the number of iterations versus γ is seemingly convex.

As the following theorem posits, this observation holds independent of the channel type and capacity under a mild assumption. Therefore, in order to design a code for the channel model given in Fig. 4.1, when γ is unknown, we force convergence of the code only for $\gamma = 0$ and $\gamma = 1$.

Theorem' 4-1 *Assuming that the distribution of the output messages of variable nodes depends only on the information they bear and the structure of the code, an LDPC code that converges on two channels with capacity C will also converge on any convex combination of those channels.*

Proof: Fig. 4.3 illustrates one iteration of density evolution on the factor graph [18] of an LDPC code. Given the aforementioned assumption, we want to show that if a specific LDPC code converges on channels 1 and 2, with LLRPDFs $f^1_{\text{ch}}(m)$ and $f^2_{\text{ch}}(m)$, where $\mathcal{I}(f^1_{\text{ch}}) = \mathcal{I}(f^2_{\text{ch}}) = C$, the code will also converge on any channel that can be seen as the convex combination of these two channels, i.e., any channel whose LLRPDF can be described as (4.1).

If the LLRPDF of the input messages to an iteration is $f_{\text{in}}(\cdot)$ and the LLRPDF of the channel is $f^j_{\text{ch}}(\cdot)$ $(j \in \{1, 2\})$, then the LLRPDF of a degree i variable node will be

$$f^j_{\text{out},i} = \bigotimes_{k=1}^{i-1} \text{CHK}\left(f_{\text{in}}, \rho(x)\right) \otimes f^j_{\text{ch}}, \tag{4.5}$$

where $\text{CHK}(f_{\text{in}}, \rho(x))$ is the pdf of the output of the check nodes in density evolution and $\otimes$ denotes the convolution operation [18].

Using (2.2), the mutual information of the output messages of degree i variable nodes will be $I^j_{\text{out},i}(I_{\text{in}}) = \mathcal{I}(f^j_{\text{out},i})$. Then, the mutual information of the output messages under the all-zero assumption will be

$$I^j_{\text{out}} = \sum_{i=2}^{\max(d_v)} \lambda_i I^j_{\text{out},i}(I_{\text{in}}). \tag{4.6}$$

In order for the code to converge on channel j, we should have

$$\sum_{i=2}^{\max(d_v)} \lambda_i I^j_{\text{out},i}(I_{\text{in}}) > I_{\text{in}}, \quad \forall I_{\text{in}} \in [C, 1). \tag{4.7}$$

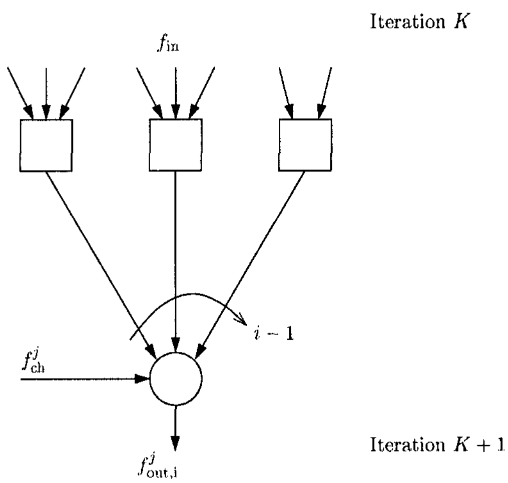


Figure 4.3: One iteration of density evolution.

Similar to (4.5), we have

$$f^{\gamma}_{\text{out},i} = \bigotimes_{k=1}^{i-1} \text{CHK}\left(f_{\text{in}}, \rho(x)\right) \otimes f^{j}_{\gamma},$$

where $f^{\gamma}_{\text{out},i}$ is the LLRPDF of a degree i variable node. Using (4.1), it follows that

$$f^{\gamma}_{\text{out},i} = \bigotimes_{k=1}^{i-1} \text{CHK}\left(f_{\text{in}}, \rho(x)\right) \otimes \sum_{j=1}^{2} \gamma_j f^{j}_{\text{ch}},$$

where $\gamma_1 = \gamma$, $\gamma_2 = 1 - \gamma$. Thus,

$$f^{\gamma}_{\text{out},i} = \sum_{j=1}^{2} \gamma_j f^{j}_{\text{out},i}. \tag{4.8}$$

Then, convergence on the combined channel is obtained because

$$\sum_{i=2}^{\max(d_v)} \lambda_i I^{\gamma}_{\text{out},i}(I_{\text{in}}) \overset{(a)}{=} \sum_{j=1}^{2} \gamma_j \sum_{i=2}^{\max(d_v)} \lambda_i I^{\gamma}_{\text{out},i}(I_{\text{in}}) \overset{(b)}{>} I_{\text{in}}, \quad \forall I_{\text{in}} \in [C, 1),$$

where (a) follows from (4.8) and (4.6), and (b) follows from (4.7).

If channel 1 has capacity C_1 and channel 2 has capacity C_2, using the above method only convergence in the region of $[\max\{C_1, C_2\}, 1)$ is obtained.

Corollary 1 *The result of Theorem 4-1 can be extended to the convex combination of N channels with equal capacity.*

Additional channels only increase the number of constraints of the linear program. Due to the very low complexity of the code-design stage, one can easily design a code with N in the order of a few hundreds. While the design complexity grows linearly with N, the number of channels for which convergence is obtained (i.e., the convex hull of N channels) grows exponentially with N.

The assumption in Theorem 4-1 is a realistic one because of the central limit theorem. That is, since LLR messages are added together in variable nodes, the distribution of messages at the output of variable nodes of a given degree is close to a Gaussian distribution. Therefore, the messages that are going into the check nodes (input to the next iteration) have a mixture of Gaussian distribution [23,31] which is uniquely defined by the code degree distribution and the input mutual information. Therefore, we expect codes that will be designed based on this theorem to have strong universal properties.

It should be emphasized here that this approach must not be mistaken with the common practice of Gaussian assumption, where all the messages (even those coming out of check nodes) are assumed Gaussian.

4.4 Identical-Capacity Channel Decomposition

In this section, we propose a channel decomposition method that expresses every given BISO channel of capacity C as a convex combination of a number of basis channels with that capacity. Using the results of the previous section, a code that works on the basis channels is expected to be suitable for all channels with capacity C.

It was stated before that BISO channels can be fully described by their LLR-PDF, or their PEPDF. With PEPDF decomposition technique, a BISO channel is spanned over BSCs with different crossover probability, $p \in [0, \frac{1}{2}]$. This decomposition is not suitable for our objective because we seek subchannels that have similar capacity. Nonetheless, we utilize this representation because it helps us to form another channel representation which captures the decomposition of the channel into identical-capacity subchannels.

From (4.2), it is understood that the channel illustrated in Fig. 4.1 can be described as a linear combination of PEPDFs of subchannels 1 and 2. The weights of this linear combination are the same as the fraction of bits passed through each subchannel. Also notice that for a BSC with capacity C, PEPDF is $g_{\mathrm{BSC}}(p) = \delta(p-\eta)$, where

$$\eta = H^{-1}(1-C), \tag{4.9}$$

and $H(p)$ is the binary entropy function, i.e., $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ for $p \in [0, \frac{1}{2}]$. Thus, when $g(p)$ is the PEPDF of a channel, the capacity of this channel can be calculated as

$$\begin{aligned} C &= 1 - \mathrm{E}\{H(p)\} \\ &= 1 - \int_0^{\frac{1}{2}} g(p)H(p)dp, \end{aligned}$$

where integration includes mass points at $p = 0$ and $\frac{1}{2}$.

4.4.1 Identical-capacity basis

Now, we focus on finding a basis, made up of channels with capacity C, that allows for expansion of every symmetric channel with capacity C. We contend that the sought basis can be a set of channels defined as follows

$$g_{x,y}(p) = \alpha(x,y)\delta(p-x) + \beta(x,y)\delta(p-y), \tag{4.10}$$

where

$$\alpha(x,y) + \beta(x,y) = 1, \tag{4.11}$$

and

$$\alpha(x,y)H(x) + \beta(x,y)H(y) = 1 - C. \tag{4.12}$$

The first condition is required for a valid PEPDF and the latter condition ensures that the capacity of each basis channel $g_{x,y}(p)$ is C.

From (4.12), it is apparent that in order for coefficients $\alpha(x,y)$ and $\beta(x,y)$ to be nonnegative, we have $\min\{x,y\} \leq \eta$ and $\max\{x,y\} \geq \eta$. Without loss of generality we assume $x \leq y$, thus $\min\{x,y\} = x$ and $\max\{x,y\} = y$. For a given capacity C, from (4.11) and (4.12), $\alpha(x, y)$ and $\beta(x, y)$ can be uniquely found as follows:

$$\alpha(x,y) = \frac{H(y) - (1-C)}{H(y) - H(x)}, \tag{4.13}$$

$$\beta(x,y) = \frac{(1-C) - H(x)}{H(y) - H(x)}. \tag{4.14}$$

From (4.10), (4.13), and (4.14), it becomes clear that $\alpha(x, y), \beta(x, y)$ and therefore $g_{x,y}(p)$ are functions of C. To simplify the notations, this dependency is not explicitly shown.

The next step is to prove that any given symmetric channel with capacity C, whose corresponding PEPDF is $g(p)$, can be expressed as a linear combination of $\{g_{x,y}(p)|(x, y) \in \mathcal{D}\}$, where $\mathcal{D} = \{(x, y) \in [0, \eta] \times [\eta, \frac{1}{2}]\}$. To this end, we find a two-dimensional pdf $q(x, y)$ which fully describes the channel according to

$$g(p) = \int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x,y) g_{x,y}(p) dy dx, \tag{4.15}$$

where η is defined in (4.9). Here, $q(x, y)$ is a density function defined over $\mathcal{D}$ which provides an alternative representation of the channel and satisfies

$$\int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x,y) dy dx = 1. \tag{4.16}$$

The next theorem shows that for any PEPDF, an equivalent description according to $q(x, y)$ exists. In this theorem, it is assumed that the PEPDF does not have a mass point at η (this simplifies the notations by removing any concern about the boundaries). In case that $g(p)$ does have a mass point at η, it can be readily separated. That is, we have

$$g(p) = a\delta(p - \eta) + (1 - a) g_1(p).$$

The first term corresponds to a BSC subchannel of capacity C and would be expressible as a mass point of height a at $x = y = \eta$ in $q(x, y)$ domain. It follows that $g_1(p) = \frac{g(p) - a\delta(p-\eta)}{1-a}$ does not have a mass point at η. Thus, one can continue the discussion with $g_1(p)$ instead of $g(p)$.

Theorem' 4-2 *Every PEPDF $g(p)$ can be written as $g(p) = \int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x, y) g_{x,y}(p) dy dx$, for some probability distribution function $q(x, y)$ defined over $\{(x, y) \in [0, \eta] \times [\eta, \frac{1}{2}]\}$.*

Proof: Any given PEPDF like $g(p)$, can be described as two halves as follows:

$$g(p) = g_\ell(p) + g_r(p),$$

where $g_\ell(p) = 0$ for $p \in (\eta, \frac{1}{2}]$ and $g_r(p) = 0$ for $p \in [0, \eta)$. Defining $q(x, y)$ as

$$q(x,y) = \frac{g_\ell(x) g_r(y)}{\beta(x,y) \int_\eta^{\frac{1}{2}} g_r(\xi) \frac{\alpha(x,\xi)}{\beta(x,\xi)} d\xi}, \tag{4.17}$$

(as it is detailed in the Appendix) one can verify that $g(p) = \int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x, y) g_{x,y}(p) dy dx$ and that $\int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x, y) dy dx = 1$. Moreover, since all the terms in the righthand side of (4.17) are nonnegative, $q(x, y) \geq 0$, meaning that $q(x, y)$ is a pdf. Detailed discussions on how to arrive at (4.17) are provided in the Appendix.

Theorem 4-2 proves that for every channel with capacity C, a $q(x, y)$ can be found according to (4.17). Thus, $\{g_{x,y}(p) | (x, y) \in \mathcal{D}\}$ indeed forms a set of basis for all channels with capacity C. Moreover, these basis channels form a minimal set in the sense that no single channel can be removed from the basis set. This is because for the basis channels themselves, only a unique representation with positive coefficient exists, i.e., for $g_{x_0,y_0}(p)$ we have only one equivalent $q(x, y) = \delta(x - x_0)\delta(y - y_0)$.

Nevertheless, there exist channels with capacity C for which more than one $q(x, y)$ representation can be found. While this seems to be inconsistent with minimality of the set of basis channels, the inconsistency is resolved if one notices that we seek convex combination of basis channels, which entails nonnegative coefficients. It should be emphasized that the inverse mapping is unique. That is, each $q(x, y)$ defines a unique channel (see (4.15)).

4.4.2 Decomposition

It was shown that any given channel with capacity C can be expressed as a continuous convex combination of basis channels with the same capacity, where the basis channels had only two terms in their PEPDF (see (4.10)).

For practical reasons, we assume the alphabet at the receiver of the channel is discrete and finite. This translates into having pmf instead of pdf. Note that all the discussions in this Section remain valid.

Having a finite alphabet also means having a finite basis, because the number of channels that satisfy (4.10) is finite. Assuming that the quantization levels are such that there are N_ℓ levels less than η and N_r levels greater than η, the cardinality of the basis channels is $N_\ell \times N_r$.

4.5 Experimental results

Given the capacity C and the quantization levels, the set of basis channels discussed in last Section can be easily obtained. We can use the basis channels to design a code with better universal properties. That is, if we assure that a designed code converges on all of these basis channels, using Theorem 4-1 we expect this code to provide good performance on all channels in the convex hull of the basis channels. The code-design process is similar to the code design formulation in Section 4.3.2.

We define the universal threshold (C^*) of a code as the minimum capacity that over all channels with $C > C^*$, convergence of the code is guaranteed. For a given code, using Theorems 4-1 and 4-2, one can effectively find C^* through a binary search, making sure that the code converges on all basis channels.

For all the examples in this Section we have used a 64-letter alphabet at the receiver (i.e., a 6-bit quantization) and a 9-bit sum-product decoder [18].

4.5.1 Check nodes

Thus far we have assumed that there is a given check degree distribution. In reality, one has to choose the check degree distribution too. The performance of LDPC codes, however, is not too sensitive to $\rho(x)$ and thus existing work usually put the focus of code optimization on $\lambda(x)$. For choosing $\rho(x)$, some suggestions and guidelines are provided in the literature, e.g., [8, 53]. For example, usually a good choice is to have two consequent degrees. This is what we do for the codes designed in this section.

With the large number of basis channels (typically more than 100), even with only two consecutive degrees in $\rho(x)$, optimization of the check degree distribution can still be cumbersome. Thus, for our simulations we use the best check degree distribution found for a code designed only for the BEC and BSC (similar to Example 1). Clearly, with a comprehensive optimization on $\rho(x)$, better universal codes can

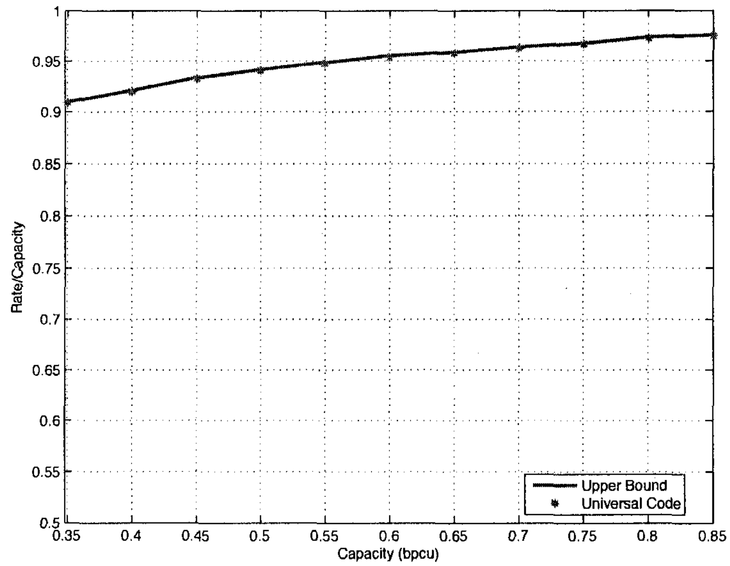


Figure 4.4: A comparison between the proposed upper bound on the achievable rates by universal LDPC codes (solid curve) and achieved rates by our universal LDPC codes (stars).

be obtained. However, the performance loss due to this simplifying assumption is minor.

4.5.2 Numerical results

We design universal LDPC codes that can work over channels with capacity $0.35 \leq C \leq 0.85$ bits per channel use (bpcu). In order to see how successful the designed codes are, we suggest an upper bound to compare our codes with. The upper bound that we consider is the rate of LDPC codes designed to work on the BEC and BSC. This code-rate defines an upper bound on the achievable rate of a universal LDPC code with similar parameters. This is because working on the BEC and BSC is a necessary (but not sufficient) condition for a universal code. This comparison is provided in Fig. 4.4, which shows that achieved rates are very close to the upper bound.

The next comparison is between a rate 0.6 code optimized for BIAWGN channel with maximum variable node degree of 100 [49], and a rate 0.6 universal code designed with the suggested method having maximum variable node of 50. Fig. 4.5 shows the decoder's message error rate obtained after 400 iterations of density evolution for these two codes over three channels (BEC, BSC, and BIAWGN). Al-

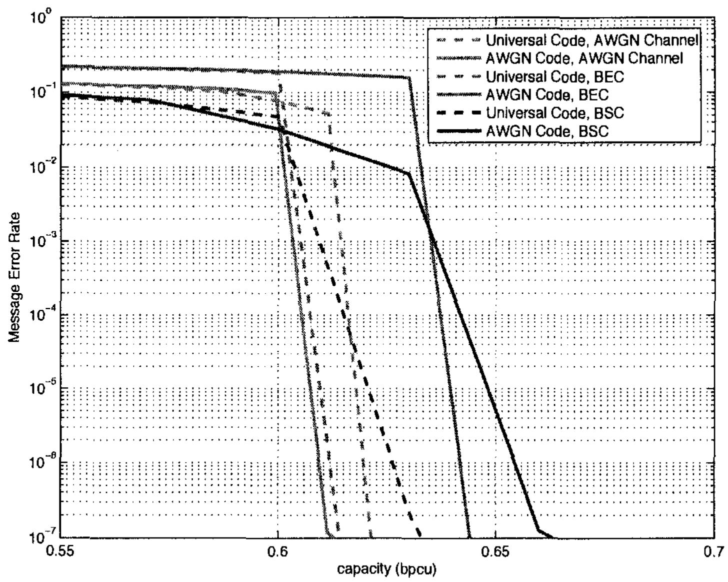


Figure 4.5: Comparison between message error rate of a rate 0.6 universal code and a rate 0.6 code designed specifically for BIAWGN on various channels. The curves are obtained by running density evolution for 400 iterations. The universal code performs almost similarly across different channels.

though the first code is slightly better on BIAWGN channels, on other channels it is outperformed by the universal code. A bit error rate comparison of these two codes, provided in Fig. 4.6, replicates this phenomenon.

The universal threshold for the BIAWGN code is $C^* = 0.662$ whereas the universal threshold of the less complex universal code is $C^* = 0.628$. Using density evolution we tested both codes on 10,000 randomly generated channels with the capacity of their universal threshold. The codes performed well on all channels.

It can be seen that the universal code has similar performance on all channels. On the other hand, the channel specific code has very good performance on the channel for which it is designed, but not viable performance on other channels.

Degree distribution for the the code designed for BIAWGN taken from [49] is, $\rho_{13} = 0.5$, $\rho_{14} = 0.5$, $\lambda_2 = 0.1499$, $\lambda_3 = 0.1621$, $\lambda_6 = 0.0224$, $\lambda_7 = 0.1764$, $\lambda_8 = 0.0077$, $\lambda_{17} = 0.1166$, $\lambda_{28} = 0.0307$, $\lambda_{29} = 0.0319$, $\lambda_{31} = 0.0438$, $\lambda_{32} = 0.0278$, $\lambda_{43} = 0.0048$, $\lambda_{100} = 0.2258$, and for the universal code is, $\rho_{12} = 0.5806$, $\rho_{13} = 0.4194$, $\lambda_2 = 0.1689$, $\lambda_3 = 0.1924$, $\lambda_6 = 0.0604$, $\lambda_7 = 0.2069$, $\lambda_{11} = 0.0763$, $\lambda_{30} = 0.0457$, $\lambda_{50} = 0.2495$.

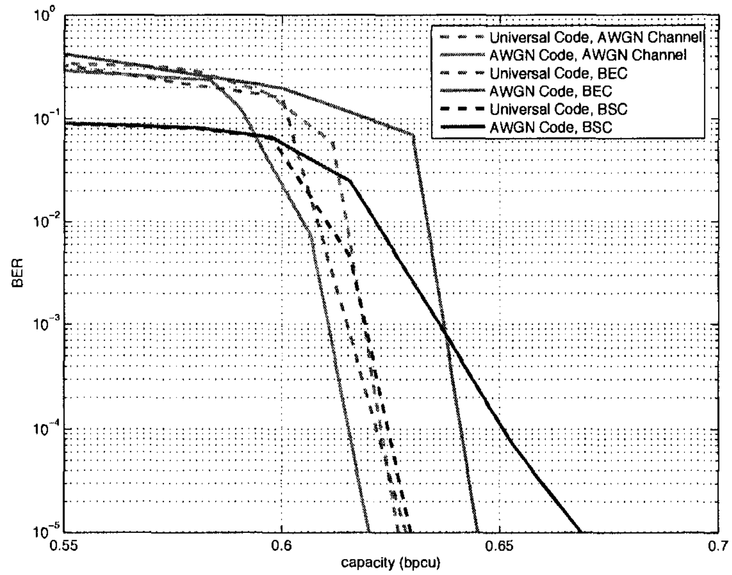


Figure 4.6: Comparison between bit error rate of a rate 0.6 universal code and a rate 0.6 code designed specifically for BIAWGN on various channels. The curves are obtained for length *100,000* randomly constructed codes.

4.6 Conclusion

Design of universal LDPC codes over BISO channels was studied. It was shown that under a reasonable assumption, codes that converge on N equal-capacity channels, also converge on the convex hull of these N channels. Therefore, we expect codes that are designed using this result to exhibit good universal properties on the specified convex hull.

In order to obtain codes with good universal properties on all BISO channels with capacity C, we proposed a channel decomposition technique which allowed for spanning any given channel with capacity C on a number of basis channels with identical capacity. We designed codes for these basis channels. As expected, these codes exhibited strong universal performance. Specifically,

1. In comparison with existing LDPC codes designed for a given channel, significant performance gain was obtained when transmission took place over various channels of equal capacity.

2. Defining a universal threshold for a code, we observed that our codes have better universal threshold compared to codes designed for specific channels.

3. We proposed an upper bound on achievable rates by universal LDPC codes and observed that our codes approach this bound very closely.

Chapter 5

Conclusion

This chapter recapitulates the contributions of this thesis and presents a number of ideas about possible future research in the related area.

5.1 Contributions

The contributions of this thesis are twofold: Firstly, code design considerations for nonuniform channels, and secondly, a method for design of universal LDPC codes.

Code Design for Nonuniform Channels

It was shown that semi-regular allotted LDPC codes can be designed via mixed integer linear programming. The proposed approach can be simplified to linear programming the semi-regularity constraint is removed.

Since allotted codes require the channel state information at the transmitter and channel knowledge at the transmitter does not change the capacity, a natural question is whether allotted codes provide any benefit over conventional LDPC codes. We argued that under optimal decoding the performance gap between allotted and conventional codes is minor. This gap, however, can be quite significant if suboptimal decoding is used or when the maximum node degree allowed in the code is small. Therefore, when channel state information is available at both the transmitter and receiver sides, it is better to use allotted LDPC codes, since they approach the capacity with smaller node degrees in the code and also because they outperform conventional codes with suboptimal decoders. We also investigated the case when neither the transmitter nor the receiver has the channel knowledge. While a capacity loss is inevitable, the reduced capacity appears to be approachable by conventional LDPC codes.

A Method for Design of Universal LDPC codes

Design of universal LDPC codes over BISO channels was studied. It was shown that under a reasonable assumption, codes that converge on N equal-capacity channels, also converge on the convex hull of these N channels. Therefore, we expect codes that are designed using this result to exhibit good universal properties on the specified convex hull.

In order to obtain codes with good universal properties on all BISO channels with capacity C, we proposed a channel decomposition technique which allowed for spanning any given channel with capacity C on a number of basis channels with identical capacity. We designed codes for these basis channels. As expected, these codes exhibit strong universal performance. In comparison with existing LDPC codes designed for a given channel, significant performance gain was obtained when transmission took place over various channels of equal capacity. Defining a universal threshold for a code, we observed that our codes have better universal threshold compared to codes designed for specific channels. We proposed an upper bound on achievable rates by universal LDPC codes and observed that our codes approach this bound very closely.

5.2 Suggestions for Future Research

Suggestions for future research are enlisted in two categories, following the main two chapters of this thesis.

Nonuniform Channels

The coding solutions suggested by this work were general in nature. However, for certain applications it seems that there can be more attractive solutions. In particular, for computer network were most of the deficiency comes from lost packages, a marriage between our coding solutions and rateless coding solutions [55, 56] seems encouraging.

Moreover, in practice, most of the channels are not memoryless. Removing the memoryless assumption, however, results is many different complications. Introducing a model that will allow for better treatment of channels with memory will result in better code design methods for such channels.

Universal Coding

Part of discussion on universal coding was based on an approximation that in density evolution, the messages coming out of check nodes depends only on the code structure and the mutual information (mutual with the all-zero code) of the messages received by the check nodes. This assumption, however close to reality, prevents the current work from claiming that the designed universal codes are analytically proven to be universal: working across all the channels with the capacity the code is designed for. Thus, a proof for Theorem 4-1 without this assumption is particularly worthy.

Another suggestion is that in the transition from an LDPC code designed for a particular channel to a universal LDPC code, there are compromises to be made in terms of, for example, the rate of the code. Also, a code might be universal across a number of channels not all channels with similar capacity. Thus, there is a need for new measures that allow for better cost-benefit analysis of different coding solutions with regard to the level of universality.

Bibliography

[1] V. Mannoni, G. Gelle, and D. Declercq, "A linear criterion to optimize irregular LDPC codes for OFDM communications," in *Proc. IEEE Vehicular Technology Conf. (VTC)*, vol. 1, 2005.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, Jul./Oct. 1948.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 1st ed. New York: Wiley, 1991.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Communications (ICC)*, Geneva, Switzerland, 1993, pp. 1064–1070.

[5] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[6] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEEE Electronics Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[7] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

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

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

[10] M. Ardakani, T. Esmailian, and F. R. Kschischang, "Near-capacity coding in multi-carrier modulation systems," *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1880–1889, Nov. 2004.

[11] E. Eleftheriou and S. Ölçer, "Low-density parity-check codes for digital subscriber lines," in *Proc. IEEE International Conference on Communications*, 2002, pp. 1752–1757.

[12] S.-Y. Chung, "On the Construction of Some Capacity-Approaching Coding Schemes," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[13] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of LDPC codes depend on the channel?" *IEEE Trans. Commun.*, vol. 54, no. 12, pp. 2129–2132, Dec. 2006.

[14] F. Peng, W. Ryan, and R. Wesel, "Surrogate-channel design of universal LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 6, 2006.

[15] T. Richardson and R. Urbanke, *Modern Coding Theory*, Oct. 2007. [Online]. Available: http://lthcwww.epfl.ch/mct/

[16] I. Sutskover, S. Shamai, and J. Ziv, "Constrained information combining: theory and applications for ldpc coded systems," *IEEE Trans. Inform. Theory*, vol. 53, no. 5, pp. 1617–1643, 2007.

[17] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University, Sweeden, 1996. [Online]. Available: http://www.it.isy.liu.se/publikationer/LIU-TEK-THESIS-440.pdf

[18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

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

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

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

[22] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[23] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.

[24] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, April 2004.

[25] H. Pishro-Nik, N. Rahnavard, and F. Fekri, "Non-uniform error correction using low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2702 - 2714, Jul. 2005.

[26] V. Mannoni, D. Declercq, and G. Gelle, "Optimized irregular Gallager codes for OFDM transmission," in *IEEE Int. Symposium on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, vol. 1, Lisbon, Portugal, Sep. 2002, pp. 222–226.

[27] A. de Baynast, A. Sabharwal, and B. Aazhang, "LDPC code design for OFDM channel: Graph connectivity and information bits positioning," in *Int. Symposium on Signals, Circuits and Systems. (SCS)*, Iasi, Romania, Jul. 2005, pp. 649–652.

[28] A. Sanaei and M. Ardakani, "Ldpc code design for nonuniform power-line channels," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, May 2007.

[29] ——, "Ldpc code design considerations for non-uniform channels," *IEEE Trans. Commun.*, submitted, Dec. 2006.

[30] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[31] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[32] I. Sason and G. Wiechman, "On achievable rates and complexity of LDPC codes over parallel channels: Bounds and applications," *IEEE Trans. Inform. Theory*, vol. 53, no. 2, pp. 580–598, Feb. 2007.

[33] A. Goldsmith, *Wireless Communications.* Cambridge, UK: Cambridge University Press, 2005.

[34] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[35] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol. 23, pp. 371–377, May 1977.

[36] M. Ardakani, B. Smith, W. Yu, and F. R. Kschischang, "Complexity-optimized LDPC codes," in *Proc. 43rd Annual Allerton Conference on Communication, Control and Computing*, Allerton House, Sep. 2005.

[37] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.

[38] R. Liu, P. Spasojevic, and E. Soljanin, "Reliable channel regions for good binary codes transmitted over parallel channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1405–1424, Apr. 2006.

[39] A. Khandekar and R. J. McEliece, "On the complexity of reliable communication on the erasure channel," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, 2001, p. 1.

[40] I. Sason and R. Urbanke, "Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1247–1256, Jun. 2004.

[41] T. Esmailian, F. R. Kschischang, and P. G. Gulak, "In-building power lines as high speed communication channels: channel characterization and a test channel ensemble," *International Journal of Communication Systems*, vol. 16, pp. 381–400, May 2003.

[42] R. D. Wesel, X. Liu, and W. Shi, "Trellis codes for periodic erasures," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 938–947, Jun. 2000.

[43] M. B. Pursley and J. S. Skinner, "Decoding strategies for turbo product codes in frequency-hop wireless communications," in *Proc. IEEE Int. Conf. Communications (ICC)*, vol. 4, May 2003, pp. 2963–2968.

[44] C. Jones, A. Matache, T. Tian, J. Villasenor, and R. Wesel, "The universality of LDPC codes on wireless channels," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, vol. 1, Oct. 2003, pp. 440–445.

[45] C. Köse and R. D. Wesel, "Universal space-time trellis codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2717–2727, Oct. 2003.

[46] X. Jin, T. Li, O. M. Collins, and T. E. Fuja, "The universality of LDPC codes on correlated fading channels with decision feedback based receiver," in *IEEE Global Telecommmn. Conf. (GLOBECOM)*, vol. 3, Nov./Dec. 2005.

[47] J. Shi and R. D. Wesel, "A study on universal codes with finite block lengths," *IEEE Trans. Inform. Theory*, vol. 53, no. 9, pp. 3066–3074, Sep. 2007.

[48] A. Sanaei and M. Ardakani, "Identical-capacity channel decomposition for design of universal ldpc codes," *IEEE Trans. Commun.*, submitted, Dec. 2007.

[49] A. Amraoui, "Ldpcopt, online code design program." [Online]. Available: http://lthcwww.epfl.ch/research/ldpcopt/

[50] I. Land, S. Huettinger, P. Hoeherand, and J. Huber, "Bounds on information combining," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 612–619, 2005.

[51] W. Yu, M. Ardakani, B. Smith, and F. Kschischang, "Complexity-optimized low-density parity-check codes for Gallager decoding algorithm b," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Sep. 2005, pp. 1488–1492.

[52] M. Ardakani and F. R. Kschischang, "Designing irregular LDPC codes using EXIT charts based on message error probability," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Lausanne, Switzerland, Jul. 2002, p. 454.

[53] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[54] M. Tuchler and J. Hagenauer, "EXIT charts and irregular codes," in *Proc. Conf. on Info. Sci. and Sys.*, Princeton, USA, Mar. 2002, pp. 748–753.

[55] M. Luby, "LT codes," in *Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, Nov. 2002, pp. 271–280.

[56] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 2033–2051, May 2006.

Appendix A

Discussions on Theorem 4-2

In this appendix, detailed discussions on how to arrive at (4.17) are provided. These discussions provide a straightforward approach to proving properties of $q(x, y)$ and clarifies some of the steps taken in the numerical results section.

Recall that $g(p)$ was described as two halves as follows:

$$g(p) = g_\ell(p) + g_r(p),$$

where $g_\ell(p) = 0$ for $p \in (\eta, \frac{1}{2}]$ and $g_r(p) = 0$ for $p \in [0, \eta)$.

First, let us assume that $g_\ell(p) = \sigma(x_1)\delta(p - x_1)$ and $g_r(p)$ consists of t mass points at $\{y_1, y_2, \ldots, y_t\} \in [\eta, \frac{1}{2}]$. This channel can be decomposed into t subchannels, i.e., $\{g_{x_1,y_i}(p)\}_{i=1}^t$ each with corresponding coefficient $\{\mu_1, \mu_2, \ldots, \mu_t\}$ satisfying $\sum_{i=1}^t \mu_i = 1$. It is easy to show that

$$\sigma(x_1) = \sum_{i=1}^{t} \mu_i \alpha(x_1, y_i),$$

$$\mu_i = \frac{g_r(y_i)}{\beta(x_1, y_i)} \quad i = 1, \ldots, t$$

and

$$q(x_1, y_i) = \delta(x - x_1)\frac{g_r(y_i)}{\beta(x_1, y_i)}$$

One can easily extend this decomposition to the continuous case by

$$q(x_1, y) = \delta(x - x_1)\frac{g_r(y)}{\beta(x_1, y)},$$

while remembering that we have

$$\sigma(x_1) = \int_\eta^{\frac{1}{2}} g_r(y)\frac{\alpha(x_1, y)}{\beta(x_1, y)} dy$$

and

$$\int_{\eta}^{\frac{1}{2}} \frac{g_r(y)}{\beta(x_1, y)} dy = 1. \tag{A.1}$$

Now we consider continuous $g_r(p)$ and s mass points at $\{x_1, x_2, \ldots, x_s\} \in [0, \eta]$ for $g_\ell(p)$. The mass points are weighted by some coefficients, let say $\{\nu_1, \nu_2, \ldots, \nu_s\}$ which satisfy $\sum_{j=1}^{s} \nu_j = 1$. Again, we can get

$$\nu_j = \frac{g_\ell(x_j)}{\sigma(x_j)} \quad j = 1, \ldots, s$$

and after extending to the continuous case we arrive at

$$\begin{aligned} 1 &= \int_0^{\eta} \frac{g_\ell(x)}{\sigma(x)} dx \\ &= \int_0^{\eta} \frac{g_\ell(x)}{\int_{\eta}^{\frac{1}{2}} g_r(y) \frac{\alpha(x,y)}{\beta(x,y)} dy} dx. \end{aligned} \tag{A.2}$$

Finally, we can derive $q(x, y)$ as

$$\begin{aligned} q(x, y) &= \int_0^{\eta} q(\xi, y) \frac{g_\ell(\xi)}{\sigma(\xi)} d\xi \\ &= \frac{g_\ell(x) g_r(y)}{\beta(x, y) \int_{\eta}^{\frac{1}{2}} g_r(\xi) \frac{\alpha(x,\xi)}{\beta(x,\xi)} d\xi}. \end{aligned}$$

In what follows, we verify that the derived $q(x, y)$ satisfies (4.16) and (4.15). Firstly, according to (A.1) and (A.2), we yield

$$\begin{aligned} &\int_0^{\eta} \int_{\eta}^{\frac{1}{2}} q(x, y) dy dx \\ &= \int_0^{\eta} \frac{g_\ell(x)}{\int_{\eta}^{\frac{1}{2}} g_r(\xi) \frac{\alpha(x,\xi)}{\beta(x,\xi)} d\xi} dx \\ &= 1. \end{aligned}$$

Secondly, using (4.10), we can write

$$\begin{aligned} &\int_{x=0}^{\eta} \int_{y=\eta}^{\frac{1}{2}} q(x, y) g_{x,y}(p) dy dx \\ &= \int_{\eta}^{\frac{1}{2}} q(p, y) \alpha(p, y) dy + \int_0^{\eta} q(x, p) \alpha(x, p) dx. \end{aligned}$$

Now, we show that the last two integrals are $g_\ell(p)$ and $g_r(p)$, respectively, because

$$
\begin{aligned}
&\int_\eta^{\frac{1}{2}} q(p,y)\alpha(p,y)dy \\
&= \int_\eta^{\frac{1}{2}} \frac{g_\ell(p)g_r(y)\alpha(p,y)}{\beta(p,y)\int_\eta^{\frac{1}{2}} g_r(\xi)\frac{\alpha(p,\xi)}{\beta(p,\xi)}d\xi} dy \\
&= \frac{g_\ell(p)}{\int_\eta^{\frac{1}{2}} g_r(\xi)\frac{\alpha(p,\xi)}{\beta(p,\xi)}d\xi} \times \int_\eta^{\frac{1}{2}} g_r(y)\frac{\alpha(p,y)}{\beta(p,y)}dy \\
&= g_\ell(p)
\end{aligned}
$$

and

$$
\begin{aligned}
&\int_0^\eta q(x,p)\alpha(x,p)dx \\
&= \int_0^\eta \frac{g_\ell(x)g_r(p)\beta(x,p)}{\beta(x,p)\int_\eta^{\frac{1}{2}} g_r(\xi)\frac{\alpha(x,\xi)}{\beta(x,\xi)}d\xi}dx \\
&= g_r(p) \times \int_0^\eta \frac{g_\ell(x)}{\int_\eta^{\frac{1}{2}} g_r(\xi)\frac{\alpha(x,\xi)}{\beta(x,\xi)}d\xi}dx \\
&= g_r(p),
\end{aligned}
$$

where the last equality results from (A.2).