LUND UNIVERSITY

Enhancing Iterative Algorithms with Spatial Coupling

Mashauri, Mgeni Makambi

2024

Document Version:
Publisher's PDF, also known as Version of record

Enhancing Iterative Algorithms with Spatial Coupling

Doctoral Dissertation

Mgeni Makambi Mashauri

LUND UNIVERSITY

Department of Electrical and Information Technology
Lund University, Lund, Sweden
October, 2024

Department of Electrical and Information Technology
Lund University
Box 118, SE-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 10pt
with the LaTeX Documentation System

Series of licentiate and doctoral theses
ISSN 1654-790X-178
ISBN: 978-91-8104-213-9 (print)
ISBN: 978-91-8104-214-6 (pdf)

Printed in Sweden by *Tryckeriet i E-huset*, Lund.
October 2024.

Kwa mke wangu mpendwa

Popular Science Summary

How can we ensure reliable communication at high data rates to meet the growing demands of modern life? How can we minimize the number of tests and thus save the cost if we want to test for a rare disease in a large population? These two questions at first glance seem to be unrelated but a closer look reveals that they have much in common and we can use similar techniques to answer both of them. We start by answering the communication question where there is a growing demand for faster data rates due to popular applications like video streaming, video calls, and online gaming. The shift towards cloud computing further expands this need for fast and reliable data transfer. We proceed to show how we can apply the tools from communication theory in what is called group testing (GT) for testing of diseases or any anomaly like detecting portions of large data which has been tampered with in data forensics or detecting faults in large computer networks.

A communication system transmits information from a source to a destination through some channel, which can be a cable or air (for wireless communication). However, channels often introduce disturbances, the most common being noise, which can make messages unclear, especially at higher data rates. One way to counteract this is to increase the transmit power, but this can lead to higher energy consumption and potential damage to equipment or harm to people. Furthermore increasing the transmit power may lead to increased interference to other users who are also communicating over the medium. An analogy for this having several people in a room with communication between pairs. If each pair were to raise their voice it would not result into better comprehension since the disturbance form other pairs will be worse. A more efficient method is to use error-correcting codes, or channel codes, which add controlled redundancy to allow the receiver to correct errors. Claude Shannon identified a limit to data rate, known as the channel capacity, within which reliable communication is possible despite noise.

Several error-correcting codes have been developed to meet modern communication needs, including convolutional codes, turbo codes, and low-density parity check (LDPC) codes. A common approach for describing these codes is through graphs, where variable nodes represent code bits, and constraint nodes represent the constraints on those bits.

Channels can also introduce other issues, like intersymbol interference (ISI), which occurs when signals sent at different times overlap at the receiver. This can happen in wireless communication when signals bounce off structures, creating delayed copies that cause interference. To address ISI, equalization is used, which aims to remove these interfering copies. Although it's tempting to address noise and ISI separately, this approach leads to sub-optimal perfor-

mance. A better solution is to use a receiver that jointly considers both code and channel, but this is complex and often impractical.

A more manageable solution is turbo equalization, where the equalizer and decoder exchange information in cycles. This method provides good performance with manageable complexity. The performance of coding systems generally has two regions: the waterfall, where the error rate drops quickly, and the error floor, where it stabilizes. To get the best performance with turbo equalization, the code and equalizer must be aligned. This has traditionally been done by designing codes, like LDPC codes, for specific channel conditions, but this approach can lead to bad error floors and inflexibility with changing conditions.

A more robust solution is to use a technique known as spatial coupling. With spatial coupling blocks of bits are interlinked to form a chain instead of being processed separately. This results in significant improvement in performance in the waterfall region without compromising the waterfall. But more importantly the performance with spatial coupling is robust even if the channel conditions changes.

The problem of group testing is to reduce the number of tests by testing items in groups instead of individual testing. The tests can be quantitative whereby each test shows the number of items which are defective or non quantitative whereby the test result is positive if at least one item is defective and negative otherwise. The test setup can be modelled by graphs thus attracting the use the tools from codes on graphs in channel coding. We show that with LDPC codes we can solve the problem of quantitative GT with low complexity and reduce the number of tests significantly. Using spatial coupling the performance of group testing is improved much more. Furthermore, we show that with spatial coupling we obtain robust performance with group testing for changing conditions in the population, particularly with changing fraction of items that are defective.

Abstract

Iterative algorithms are becoming more common in modern systems. This ranges from algorithms for communication systems receivers, machine learning, group testing, and various computation problems. The success of these algorithms lies in the ability to simplify computation by breaking down the system into components and exchanging messages on graphs. The graph has the components as nodes and connections between them as edges. This separation is needed since attempting to solve the problem without dividing it into parts results into an optimal solution, the joint maximum *a posterior* (MAP) solution, but the computational complexity is prohibitive.

With the systems divided into separate parts it often seems reasonable to use the best component for each part to achieve good performance. This, however, results into degraded performance compared to the optimal overall solution. To get improved performance the components have to exchange information iteratively in a number of cycles a process known as belief propagation (BP). This principle has been applied with much success in various areas such as the design of turbo codes and low density parity-check (LDPC) codes for reliable communication. Other examples include iterative receivers for cancelling intersymbol interference (ISI) and better performance of modulation and coding in coded modulation.

Choosing component codes for communication systems with iterative systems is often a process which involves compromises. For example, if one chooses a strong code to work with a particular detector, the resulting performance in the waterfall region becomes poor but the error floor is improved whereas choosing a weak code results in improved waterfall region but poor error floor. One can also optimize the code, for example, by tuning the degree distribution of LDPC codes to achieve good performance but the optimization introduces weak components that compromise the error floor. Furthermore, the optimization can work well for a given set of channel conditions, but the optimized code may not work well if the conditions are changed.

These problems are a result of the fact that we have limitations from two aspects. First, we are limited by the component (e.g. codes or detectors) choice which sets the limit on the MAP threshold. A strong code will then have a good MAP threshold and good error floor wheres a weak code will have a bad MAP threshold and bad error floor. It is important to note that the MAP threshold is the best we can do with the choice of the components but it can still be away from the ultimate information theoretical limit of the system (this corresponds to the capacity in communication systems for example). A second limitation comes from the decoding algorithm. The BP algorithm is not globally optimal for most graph used, thus setting a limit which is termed the BP threshold. A

strong code has then a bad BP threshold whereas a weak code has a better BP threshold.

This thesis focuses on improving the performance of iterative algorithms by tackling the limitations highlighted. We propose improved algorithms and, more importantly, we apply the concept of spatial coupling to improve the performance and robustness of the systems. We do this in two parts.

In the first part we apply the concept on channels with ISI showing that we can obtain robust performance with changing channel conditions and changing detector type. We propose three schemes of coupling and compute the BP and MAP thresholds as well as perform finite length simulations.

In the second part, we investigate non-adaptive quantitative group testing using sparse graphs. We propose improvements of the algorithms and show that with spatial coupling we can obtain improved and robust performance.

Contribution Statement

This doctoral thesis concludes my work as a Ph.D. student, and is comprised of two main parts. The first part gives an overview of the research field in which I have been working during my Ph.D. studies and a brief summary of my work. The second part is composed of the following six papers (The papers are slightly reformatted to fit within the overall thesis structure):

1. M. M. Mashauri and M. Lentmaier, "Spatial Coupling in Turbo Equalization," *in Proc. IEEE Global Telecomm. Conf. (GLOBECOM),* Taipei, Taiwan, Dec. 2020.

2. M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "On the Universality of Spatially Coupled LDPC Codes over Intersymbol Interference Channels," in *Proc. IEEE Information Theory Workshop (ITW)*, Kanazawa, Japan, Oct. 2021.

3. M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "Robust Performance over Changing Intersymbol Interference Channels by Spatial Coupling," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Seoul, Korea, May 2022.

4. M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "Low-Density Parity-Check Codes and Spatial Coupling for Quantitative Group Testing," in *Proc. IEEE Intl. Sympos. Inf. Theory (ISIT)*, Taipei, Taiwan, June 2023.

5. M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "Spatially Coupled LDPC and GLDPC Codes for Quantitative Group Testing," To be submitted to IEEE Transactions on Information Theory.

6. M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet ," Submitted to IEEE Communication Letters, 2024 (under review).

In Paper 1, I investigated the application of spatial coupling in turbo equalization for channels with intersymbol interference (ISI). I did finite length simulations for three different schemes of introducing coupling in turbo equalization. The schemes are coupling at the input of the channel, coupling the code and coupling both the code and at the input of the channel. I also derived the extrinsic information transfer (EXIT) chart thresholds for the ISI channels with additive white Gaussian noise (AWGN). I did the writing with the help of my supervisor.

In Paper 2, I investigated the universality of spatial coupling for general ISI channels for both AWGN and erasures. To do this, I derived an exact input/output transfer function for a general ISI channel and set up the environment for running density evolution (DE), computing MAP thresholds and finite length simulations. I also did the main part in writing with the help of my supervisors.

In Paper 3, I extended the work in Paper 2 by looking at the practical implications of the universality of spatial coupling highlighted in Paper 2. This was done by comparing spatial coupling with classical code designs for changing channels and detectors. I computed the thresholds, ran the finite length simulations and did the main part in writing with the help of my supervisors.

In Paper 4, I proposed an LDPC based scheme for non-adaptive quantitative group testing and derived the corresponding DE equations. I also applied spatial coupling to the proposed scheme and the existing scheme based on GLDPC. I derived the equations, ran DE and finite length simulations and did the main part in writing with the help of my supervisors.

In Paper 5, I provided a proof of threshold saturation for quantitative group testing with LDPC codes. This was done by first reformulating the DE recursions developed in Paper 4 and proving that the recursions satisfy the vector admissible system requirements. I then computed the potential thresholds for various regular LDPC codes and compared with the coupled thresholds which as expected did match. Furthermore, I did finite length simulations to compare spatially coupled LDPC codes with their uncoupled version and the generalized LDPC (GLDPC) construction. Furthermore, I proved threshold saturation for the GLDPC based group testing by showing that the recursion satisfied a scalar admissible system. I also provided additional improvements on the GLDPC decoder using concepts learned from the decoder for LDPC codes. With the help of my supervisors I did the main writing of the paper.

In Paper 6, I proposed a novel scheme for quantitative group testing based on LDPC codes. The scheme involved introducing hidden nodes by bundling together items into groups of size q. This introduces constraints in the design of the test matrix. The bundling does not affect the testing procedure but can be utilized by the decoder to improve the decoding performance with limited increase in complexity. I derived the DE equations and computed the thresholds. I also ran finite length simulation of the scheme comparing it with the conventional LDPC proposed in Paper 4 and the GLDPC scheme. I did the main writing with the help of my supervisors.

Other Contributions

The following paper is a conference version of Paper 5 but is not included in this dissertation.

- M. M. Mashauri, A. G. i Amat, and M. Lentmaier, 'Threshold Saturation for Quantitative Group Testing with Low-Density Parity-Check Codes," in *Proc. IEEE Intl. Sympos. Inf. Theory (ISIT)*, Athens, Greece, July 2024.

The work done during this PhD has been supported by the Swedish Research Counsel (grant #2017-04370.) and the Excellence Center at Linköping-Lund in Information Technology (ELLIIT).

Acknowledgments

My PhD journey have been both exciting and challenging. I would like to express my sincere thanks to all people who have been there to give me the support I needed.

First and foremost, I would like to extend my sincere gratitude to my supervisor Michael Lentmaier. This accomplishment would not have been possible without your unwavering belief in me and your tremendous support throughout this journey. I will always remember this experience as one of learning to ask questions, finding answers, and realizing that every answer brings even more questions.

I would also like to thank Alexandre Graell i Amat for your invaluable support and suggestions, which have greatly contributed to improving my work. Additionally, thank you to Ove Edfors for your encouragement and guidance throughout my PhD studies.

I am incredibly grateful to all my colleagues at EIT, particularly those in the Communication division, for your friendship, support, and the wonderful memories we have shared.

My sincere gratitude goes to my father, Simon, and my mother, Donatila, for their support and understanding throughout my educational journey, which has enabled me to reach this point.

I also want to thank my children, Samweli, Lidia and Klementina, for their patience and for always being there for me. My thanks extend to all my family members and friends who have supported me in countless ways.

Finally, I would like to express my deepest gratitude to my wife, Victoria. Your immense support, love, understanding, and prayers have made this journey possible. *Asante sana*! *Mungu awabariki sana!*

Mgeni Makambi Mashauri

List of Acronyms

5G	Fifth Generation
APP	A-Posteriori Probability
AWGN	Additive White Gaussian Noise
BCH	Bose–Chaudhuri–Hocquenghem
BCJR	Bahl-Cocke-Jelinek-Raviv
BEC	Binary Erasure Channel
BER	Bit Error Rate
BMD	Bounded Minimum Distance
BP	Belief Propagation
CC	Convolutional Code
cdf	Cumulative Distribution Function
CN	Check/Constraint Node
dB	Decibel
DE	Density Evolution
DSD	Double-Sided Decoder
DVB-S	Digital Video Broadcasting Satellite
EXIT	Extrinsic Information Transfer
FTN	Faster-Than-Nyquist
GEXIT	Generalized Extrinsic Information Transfer
GF	Galois Field
GLDPC	Generalized Low-Density Parity-Check
GT	Group Testing
ISI	Intersymbol Interference
LDD	Low-Density Decoder

LDPC	Low-Density Parity-Check
LLR	Log-Likelihood Ratio
LMMSE	Linear Minimum Mean Square Error
LTE	Long Term Evolution
MAP	Maximum A-Posteriori
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
PCC	Parallel Concatenated Code
pdf	Probability Density Function
pmf	Probability Mass Function
RV	Random Variable
SCC	Serially Concatenated Code
SC-LDPC	Spatially Coupled Low-Density Parity-Check
SC-PCC	Spatially Coupled Parallel Concatenated Code
SC-SCC	Spatially Coupled Serially Concatenated Code
SIR	Symmetric Information Rate
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SSD	Single-Sided Decoder
VN	Variable Node
WiMAx	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Contents

Part I

Overview of Research Field

Chapter 1

Introduction

1.1 Channels with Intersymbol Interference

Communication has become an indispensable part of modern life. We have witnessed the growth of mobile communication throughout the world which was initially mainly for voice and short messages. The explosion of the Internet and its integration in the mobile network have pushed the demand for reliable communication at higher data rates.

Reliable communication is a challenge since the channel, the medium through which information propagates from the sender to the receiver, can corrupt the signal carrying the information. One common source of corruption is noise, which when added to the signal can corrupt the signal making it unintelligible to the receiver, especially when the signal has been attenuated after traveling a long distance. A quick fix to this would be to increase the transmit power such that the signal is very strong when it reaches the receiver. But this implies higher energy consumption, and there is a limit to the amount of energy the electronics in the transmitter can handle without being damaged. More power would also result in increased interference to other users. The interference could be worse if each user were to increase their transmit power.

With limited energy resources, what is then the limit to the data rate at which we can still communicate reliably? How can this be done? The first question was answered by Claude Shannon in 1948 [1] when he showed that for given channel conditions like the ratio of signal power to noise power, we can communicate reliably if the rate is below what is called the capacity of the channel. An alternative formulation of of Shannon's limit provides the minimum signal power required for almost error free communication for given rate. This can be achieved by using error-correcting codes. This involves introducing some controlled redundancy to the information bits, which can be exploited by the decoder at the receiver to correct errors which might have happened in the transmission process.

The performance of a channel coding scheme is limited by the decoding algorithm used at the receiver. The best possible performance in terms the minimum signal power for vanishing bit error rate (BER) is obtained by using the maximum a posterior (MAP). The limit of the MAP receiver (MAP threshold) can be away from the capacity limit. But MAP receivers for coded systems

are prohibitively complex thus necessitating the use of sub-optimal algorithms. The most common being the belief propagation (BP). The BP receiver has a limit (BP threshold) worse than the MAP threshold. A code with good MAP threshold usually has a poor BP threshold and a code with good BP threshold has a bad MAP thresholds. In classical code design codes are usually designed to have good BP thresholds. This however, leads to codes with poor MAP threshold thus failing to achieve the capacity limit. This dilemma, illustrated in Fig. 1.1, is addressed in the thesis.

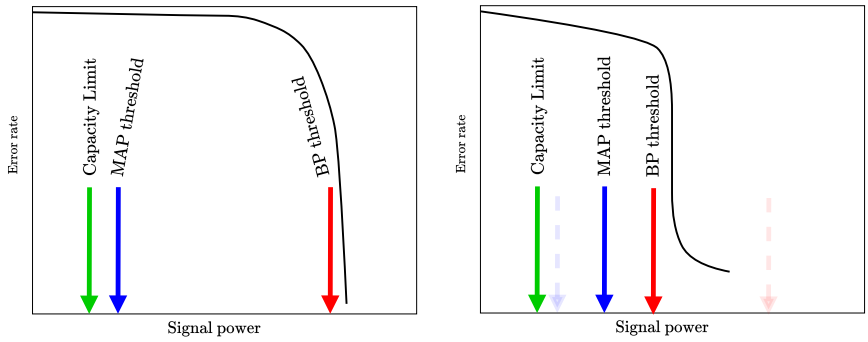


Figure 1.1: An illustration of the dilemma between the MAP threshold and BP threshold (a) A code with good MAP threshold (b) A code with good BP threshold but poor MAP threshold (The limits of the code in (a) are shown in (b) in faint dotted arrows for comparison).

A simple example of an error correcting code is the repetition code which repeats the information bits several times. If some bits are corrupted the decoder can recover the original bit by making a majority vote on the received bits. This simple repetition however does not result in efficient error correcting codes as the achievable data rates can be very far from the capacity. More elaborate codes have been developed for better error correction [2]. Examples include algebraic codes like Reed-Solomon codes and BCH codes which can better correct errors that occur in blocks. Other codes include convolutional codes that can be encoded and decoded in a stream, making them suitable for streaming applications. Concatenating two codes often results in stronger codes. Thus, for example, for digital video broadcasting standard for satellite (DVB-S) and space communication, Reed-Solomon codes are concatenated by convolutional codes to get a better performance.

A major breakthrough in channel coding came with the development of turbo codes in the early 90s. These codes could achieve almost error-free communication very close to the capacity limit [3]. The key to their performance was the use of two convolutional codes in parallel which shared the input bits with the decoders exchanging soft information in a number of iterations. One application which benefited from this was the 3G mobile communication. Variants of codes based on the principle have been developed which are collectively known as turbo-like codes [4]. It was noted that low-density parity-check (LDPC) codes, which were invented much earlier [5] but were not developed to be practical at that time, were also based on a similar principle. They were later developed and could provide similar to better performance [6, 7], leading them to be widely

used in 4G and 5G mobile communications. Both turbo-like codes and LDPC codes are termed as codes on graphs. The idea behind codes on graphs is having nodes of a graph which exchange information on a number of iterations in the decoding process in what is called belief propagation (BP) [8]. The concept of codes on graphs has made a big impact in the design and analysis of modern coding theory.

Apart from noise, the transmitted signal may be distorted by the overlap with other signals transmitted in nearby time slots. This phenomenon is called intersymbol interference (ISI). A simple example of the phenomenon can be described by wireless transmission whereby a signal might take more than one path to reach the receiver. For example, we might have the direct path and the path taken by bouncing off buildings in a city. If the transmission rate is increased, the spacing between separate symbols is reduced which might lead to overlap of the current signal with signals which were sent in previous time slot but took a longer path. Nyquist provided the formal criteria which the transmission has to fulfill in order to avoid ISI. In general the Nyquist criteria for ISI puts a limit for the transmission rate for a limited bandwidth [9].

The effect of ISI could however be mitigated by employing an equalizer at the receiver. The equalizer cancels or rather attempts to cancel the effect of the symbols from neighbouring time slots from the current signal. Since the data are also protected by a channel code, the equalizer passes its output to the decoder for error correction. It is tempting to treat these two components as separate entities, but doing that may result in poor performance when compared to the capacity of the channel. An optimal receiver would be a joint MAP receiver which would consider both the code and channel together. However, the complexity of such a receiver is prohibitive. A common solutions is to let the equalizer and the decoder exchange information iteratively in a number of iterations. Such a scheme is known as turbo equalization [10, 11] since it uses the same principle as turbo codes.

In classical code design, choosing codes to work with an equalizer in turbo equalization often results in some trade-offs between the performance in the waterfall region (the region with low SNR where the bit-error rate (BER) falls rapidly) and the error floor region (the region at higher SNR where the BER remains relatively constant). If a code which is strong in the sense that it has good performance in AWGN channel is used, the waterfall performance becomes poor but the error floor becomes good. On the other hand, if a weak code is used the waterfall performance becomes good but the error floor becomes poor. This phenomenon is connected to the fact that the BP algorithm is not optimal thus limiting the performance of iterative receivers away from the MAP threshold. The stronger code has a better MAP threshold than the weaker code but might have a poorer BP threshold compared to the weak code. If we were to use a joint MAP receiver, we would have the strong code perform better both in terms of waterfall and error floor. A common solution in classical code design is to optimize the code to fit with a particular channel and equalizer to have performance close to the capacity limit [12]. The drawback with this approach is that we need to re-optimize the code if the channel would to change making it impractical in varying channel conditions. A relatively modern concept of spatial coupling provides a solution to some of the challenges with classical code design for iterative receivers.

With spatial coupling, blocks of data are interconnected to form a chain as

opposed to encoding each block separately as in classical code design. This was first studied for LDPC codes [13]. It was observed that the performance in the waterfall region of the spatially coupled LDPC codes where improved considerably without compromising the error floor. This was later shown experimentally and formally proved to be a result of a phenomenon known as threshold saturation whereby the BP threshold of the spatially coupled code approaches the MAP threshold of the underlying uncoupled ensemble with BP decoding [14,15]. This opens up a new paradigm in code design whereby the MAP threshold is the target. We can thus use strong codes and obtain good performance in the waterfall and error floor regions with BP decoding [16]. Furthermore, spatial coupling has been shown to exhibit universality whereby the same code can achieve the capacity under different conditions [17, 18].

The purpose of this thesis when studying channels with intersymbol interference is to show that spatial coupling could be employed in turbo equalization to remove the trade-offs between the performance in the waterfall and error floor regions. We also demonstrate with concrete examples how universality of spatial coupling results in robust performance in situations where the channel might be changing. The thesis includes three papers on the topic, Paper I, II and III. In Paper I we introduce spatial coupling in turbo equalization in three different ways demonstrating that the trade-off between waterfall and error-floor can be avoided. In Paper II we investigated the universality of spatial coupling for channels with ISI for both erasures and AWGN. In Paper III we demonstrated that with spatial coupling the performance we get robust performance with changing channels as well as using sup-optimal decoders.

1.2 Group Testing with Sparse Graphs

Consider a large population of items with very few of them being "defective". This could for example be a population of people with a small fraction of individuals affected by a rare disease. The purpose of group testing (GT) is to identify the set of items which are defective with minimum number of tests by testing the items in groups instead of performing individual tests. This was first introduced by Dorfman in 1943 to test for syphilis among military draftees for World War II [19]. It results in significant saving in the number of tests thus reducing cost especially when the number of items which are defective is very small compared to the population size. Group testing has been applied in many fields. Examples include DNA sequencing [20] to identify rare traits, fault searching in electrical and computer networks [21], data forensics [22, 23] and multi-access communication in wireless communication [24, 25].

The design of the tests can either be adaptive or non-adaptive. With adaptive GT the tests are carried out in stages whereby the test design at each stage depends on the outcome of the tests in previous stages. With non-adaptive GT on the other hand, all the tests are designed beforehand. In general, adaptive GT uses less tests than non-adaptive GT but non-adaptive GT is attractive for many applications because the tests can be run in parallel thus saving testing time.

GT can also be classified into quantitative versus non-quantitative depending on the outcome of the test results. With quantitative GT the results of each test indicates how many items participating in the test are defective. With

non-quantitative GT, however, the test result indicates whether at least one item in the test is defective. That is, the test is positive if at least one item in the test is defective and negative if all the items are non-defective. Several variations exist between the two extremes such as semi quantitative GT testing which a variation of quantitative GT with the test results indicating a range of defective items in the test. Another variation includes threshold GT whereby the test is positive if the number of defective items is above a certain threshold and negative if the number is below another (possibly different) threshold. This thesis focuses on non-adaptive quantitative GT.

The test assignment can be modelled by a matrix with the columns representing the items and the rows the tests. If an item participates in a given test the entry in the test matrix corresponding to the intersection of the item and test is one and zero otherwise. This setting has similarities with channel codes where the test matrix corresponds to the parity-check matrix of a code and the test results as the syndrome. We can thus apply some of the tools in coding theory particularly codes on graphs to design and analyse the performance of GT. The corresponding graph in GT has the tests as the constraint nodes and the items as the variable nodes. For example in non-quantitative GT the algorithm SAFFRON and its variations [26, 27] uses sparse graph codes to design efficient tests with low complexity decoding. The concept was extended to quantitative GT in [28] whereby generalized LDPC (GLDPC) codes where used with t-error correcting BCH codes as the component codes.

The performance of the GLDPC codes is very good for low values of t such as $t = 1$ and $t = 2$ but degrades substantially when t is increased to higher values. This phenomenon is a result of that the increased error correcting capability comes at a cost of increasing number of tests which outweighs the benefit. Furthermore, the test design has to be optimized depending on the range of number of defective items in the population. That is a test design can be optimal when the fraction of defective items in the population in a given range but when the number of defective items is increased or decreased the design might perform poorly requiring a new design.

The purpose of this thesis when it comes to group testing is to examine closely the design of test using sparse graphs and propose more efficient schemes. We do this by proposing new schemes based on LDPC codes and improving the existing schemes. Furthermore we introduce spatial coupling in quantitative GT to improve the performance and, more importantly, to obtain robust performance for a wider range of scenarios. The thesis includes three papers on the subject, Paper IV, V, and VI. In Paper IV we introduce the proposed scheme with LDPC codes. We also introduce spatial coupling to both schemes and derive the corresponding density evolution equations. In Paper V we prove that threshold saturation occurs in quantitative GT for both the GLDPC and LDPC scheme. We also propose an improved decoders for the GLDPC scheme. In Paper VI we propose an improvement on the scheme with LDPC codes by introducing hidden nodes in the graph thus enabling the use of non-binary alphabet without changing the test operations.

1.3 Thesis Outline

This thesis is written in the paper collection format and consists of two main parts. The first part consists of an overview of the research field whereas the second part contains the research contributions in form of a selection of papers which covers two areas, one on channels with ISI and the other on group testing. The first part is divided into five chapters. Chapter 2 contains brief explanations of various fundamental concepts used in the thesis. In Chapter 3, concepts pertaining to iterative algorithms on graphs are presented. Several graph-based codes are presented showing their graphical structures and how message passing is done. Various ways of computing the decoding thresholds such as density evolution are also presented for the Gaussian and erasure channel as well as models for channel with ISI. The concept of spatial coupling is introduced in this chapter as well. Chapter 4 presents various basic concepts about group testing. We introduce graph-based GT focusing on quantitative GT. Finally, in Chapter 5, a summary of the included papers is presented together with the main conclusions of the thesis as well as suggestions of future research areas.

Chapter 2

Preliminaries

2.1 Discrete Communication Model

A simplified channel model is shown in Fig. 2.1. A block of information bits $\boldsymbol{u}$ form a data source is input to the channel encoder, which adds controlled redundancy to produce the coded bits $\boldsymbol{v}$. The modulator takes the coded bits, q at a time, and outputs the modulated symbols $\boldsymbol{x}$. The modulated symbols are

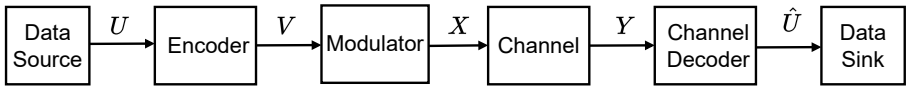


Figure 2.1: Block diagram showing a simplified discrete communication model.

passed through a channel characterized by the transition probability $P_{Y|X}(\boldsymbol{y}|\boldsymbol{x})$. We use an upper case letter to denote a random variable (RV) and a lower case one for its realization.

A channel is called memoryless if for a block of transmitted signal $\boldsymbol{x}$ and a block of received symbol $\boldsymbol{y}$ the conditional probability $\mathrm{P}_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})$ can be factored as

$$\mathrm{P}_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_i \mathrm{P}_{Y_i|X_i}(y_i|x_i)\,. \tag{2.1}$$

This means that the output of the channel at time i depends only on the input x_i at the equivalent input time. This is in contrast to channels with memory, where the output at a particular time is affected by inputs at other times as well.

At the receiver the channel decoder uses the received signal $\boldsymbol{y}$ to produce a best possible guess of u denoted as $\hat{u}$.

2.1.1 Channel with Additive White Gaussian Noise

For an AWGN channel, the output of the channel is corrupted by additive noise which is Gaussian distributed with zero mean and variance σ^2. That is, if the input X_i is real, the output Y_i is given as

$$Y_i = X_i + W_i\,,$$

where $W_i \sim \mathcal{N}(0, \sigma^2)$. We thus have

$$\mathrm{P}_{Y_i|X_i}(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y_i - X_i)^2}{2\sigma^2}\right).$$

The channel quality for this channel is usually given as the ratio $\frac{E_\mathrm{b}}{N_0}$ of the energy per bit E_b to the noise spectral density N_0 related to σ as $N_0 = 2\sigma^2$. When the input X is binary the channel is referred as binary AWGN (BI-AWGN).

2.1.2 Binary Erasure Channel

The binary erasure channel is a simple channel model which either erases the input symbol completely with probability ε or passes through with probability $1-\varepsilon$. The model is shown in Fig. 2.2. The BEC is a very simple channel model

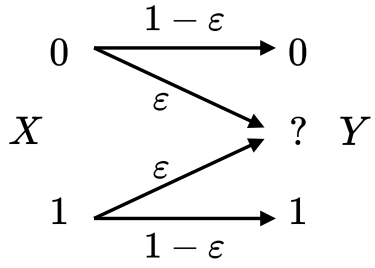


Figure 2.2: Binary erasure channel model.

which simplifies analysis using mathematical tools. For this reason it is widely used as tool to gain insight into behaviour of various communication concepts with some of the results giving close approximation to applications with more complex model like the BI-AWGN.

2.2 Information Measures

In this section we introduce information measures commonly used in analysing communication systems.

2.2.1 Entropy

Entropy is a measure of uncertainty about a random variable. For a discrete RV X, which takes values in the set $\mathcal{X}$ with probability mass function (pmf), $P_X(x)$, its entropy $H(X)$ is given as

$$H(X) = -\sum_{x\in\mathcal{X}} P_X(x) \log_2 \left(P_X(x)\right) .$$

If X is a continuous RV with a support set $\mathcal{S}$ the entropy is computed as

$$H(X) = -\int_{\mathcal{S}} f_X(x) \log_2(f_X(x)) \, ,$$

where$f_X(x)$ is the probability density function. It is common to describe the entropy of a continuous RV as differential entropy.

The uncertainty about X provided that another RV Y is known is called conditional entropy and is given as

$$\begin{aligned} H(X|Y) &= \sum_{y\in\mathcal{Y}} P_Y(y) H(X|Y=y) \\ &= \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{x\in\mathcal{X}} P_{X|Y}(x|y) \log_2 \left(P_{X|Y}(x|y)\right) . \end{aligned}$$

The conditional entropy is always less than or equal to the unconditional entropy since having additional information can reduce the uncertainty but cannot increase uncertainty.

2.2.2 Mutual Information

The reduction in uncertainty about a RV X as a result of the knowledge of another RV Y is quantified by the mutual information between X and Y. That is

$$I(X;Y) = H(X) - H(X|Y) .$$

The mutual information is symmetrical meaning that $I(X;Y) = I(Y;X)$.

2.2.3 Channel Capacity

The maximum rate we could have and still maintain error free communication is known as channel capacity C. This limit is given by the maximum mutual information between the received signal Y and the transmitted signal X. That is

$$C = \max_{P_X(x)} I(X;Y) .$$

The maximum is taken over all possible distributions $P_X(x)$ of the transmitted signal. It is, however, common to consider the capacity when with the input constrained to some discrete alphabet with a given distribution. This is often termed as constrained capacity or maximum information rate.

2.3 Galois Fields

In this section we briefly introduce the concept of fields and Galois fields (or finite fields). These concepts are useful in describing channels codes especially algebraic codes. Most of the materials in the section are based on [29].

Fields

A field is a set $\mathbb{F}$ of elements together with two binary operators "$+$" and "$\cdot$" that satisfy the following properties:

1. $\mathbb{F}$ is a commutative (abelian) group with respect to "$+$"
2. $\mathbb{F}$ is an abelian group with respect to "$\cdot$"
3. Distributively: $\forall a, b, c \in \mathbb{F}\ a \cdot (b + c) = a \cdot b + a \cdot c$.

Abelian Group

An abelian group is a set of elements $\mathbb{G}$ together with an operator "$*$" which satisfy the commutative law [30]. It also called a commutative group. In detail an abelian group satisfies the following properties:

1. Closure: $\forall a, b \in \mathbb{G} \quad a * b \in \mathbb{G}$
2. Associativity: $\forall a, b, c \in \mathbb{G} \quad a * (b * c) = (a * b) * c$
3. Identity element: $\forall a \in \mathbb{G} \exists e \in \mathbb{G} : \quad a * e = a$
4. Inverse: $\forall a \in \mathbb{G} \exists b \in \mathbb{G} : \quad a * b = e \implies b = a^{-1}$
5. Commutativity: $\forall a, b \in \mathbb{G} \quad a * b = b * a\,.$

Galois Fields

A Galois field is a field with finite number of elements. A Galois field with q elements is denoted as GF(q) or $\mathbb{F}_q$. If q is prime the field is called a **prime field**. The order of an element a in a Galois field is the smallest integer k such that $a^k = 1$. An element whose order equals $q - 1$ is called a primitive element and is denoted as α.

For example GF(2) is a prime field with elements $\{0, 1\}$ with the operators "$+$" and "$\cdot$" corresponding to addition and multiplication modulo 2.

Extension Fields

An extension field is a Galois field where the number of elements $q = p^m$, where p is a prime number and m is an integer. The non-zero elements of a Galois field can be represented by powers the primitive element α. For example for a GF(2^m) we have

$$\mathbb{F}_{2^m} \setminus 0 = \left\{1, \alpha, \alpha^2, \ldots \alpha^{2^m-2}\right\}\,.$$

The elements can also be represented as polynomials in α as

$$a_0 + a_1 \cdot \alpha + a_2 \cdot \alpha^2 \cdots + a_{m-1} \cdot \alpha^{m-1}\,,$$

where $a_i \in \mathbb{F}_p$. The corresponding binary vector representation is given as

$$(a_0, a_1, a_2 \cdots, a_{m-1})\,.$$

Extension fields with $p = 2$ can be efficiently implemented due to their binary vector representation.

2.4 Channel Codes

2.4.1 Block Codes

Block codes are defined by a mapping of a block of K information bits to a block of N codeword bits with $N > K$. The ratio $R = \frac{K}{N}$ is called the rate of the code. A code is called linear if it contains the all-zero codeword and every

linear combination of codewords is also a codeword. In other words, a linear block code $\mathcal{C}(N, K)$ is a linear subspace of dimension K of the vector space $\mathbb{F}_2^N$. The number of non-zero elements in a codeword is called the Hamming weight of the codeword. The number of positions by which two codewords differ is called the Hamming distance between the two codewords. The minimum of these distances is called the minimum distance of the code. By linearity the minimum distance is equal to the minimum weight of the non-zero codewords. Linear codes are widely used because they can be encoded and decoded with higher efficient algorithms. The encoding of linear block codes can be described using an $K \times N$ generator matrix $\boldsymbol{G}$ which contains K linearly independent codewords. The parity-check matrix $\boldsymbol{H}$ of a linear code $\mathcal{C}$ is an $(N-K) \times N$ matrix such that every codeword $\boldsymbol{v}$ of $\mathcal{C}$ satisfies the constraint

$$\boldsymbol{v}\boldsymbol{H}^{\mathsf{T}} = \boldsymbol{0}\,.$$

This means that $\boldsymbol{H}$ is a matrix with elements in $\mathbb{F}_2$ (in general for a $q-$ary code the elements are members of $\mathbb{F}_q$) such that its null space is $\mathcal{C}$. The code $\mathcal{C}_d$ whose generator matrix is the parity-check matrix $\boldsymbol{H}$ is called a dual code of $\mathcal{C}$. For example a $(K+1, K)$ single parity-check code is a dual code of $(K+1, 1)$ repetition code.

2.4.2 Convolutional Codes

Convolutional codes (CCs) are characterized by the property that they can be encoded continuously on a stream such that the output bits at a particular time slot are a function of not only the current input bits but also of input bits in previous time slots. The span of time over which bits can still affect the current output is called the memory of the code m. More formally consider the sequences of information bits $\boldsymbol{u}$ and code bits $\boldsymbol{v}$ where

$$\begin{aligned}\boldsymbol{u} &= (\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_t, \dots)\ ,\\ \boldsymbol{v} &= (\boldsymbol{v}_0, \boldsymbol{v}_1, \dots, \boldsymbol{v}_t, \dots)\ .\end{aligned}$$

A CC encoder accepts a block of k information bits, $\boldsymbol{u}_t = (u_t^{(1)}, \dots, u_t^{(k)})$ which is encoded to produce a block of n code symbols, $\boldsymbol{v}_t = (v_t^{(1)}, \dots, v_t^{(n)})$. The code bits $\boldsymbol{v}_t$ are a function of information blocks, $\boldsymbol{u}_{t'}$ for $t' = t-m, \dots, t$. That is

$$v_t^{(j)} = \sum_{i=1}^{k} \sum_{l=0}^{m} u_{t-l}^{(i)} g_{i,l}^{(j)}\ ,$$

where $g_{i,l}^{(j)}$ equal 1 if $v_t^{(j)}$ depends on $u_{t-l}^{(i)}$, otherwise $g_{i,l}^{(j)} = 0$. The vector $\boldsymbol{g}_i^{(j)} = \left(g_{i,0}^{(j)}, g_{i,1}^{(j)}, \dots, g_{i,m}^{(j)}\right)$ is called the generator vector for the input i and output j. We can thus express the encoder as

$$\boldsymbol{v}^{(j)} = \boldsymbol{u}^{(1)} \circledast \boldsymbol{g}_1^{(j)} + \boldsymbol{u}^{(2)} \circledast \boldsymbol{g}_2^{(j)} + \dots + \boldsymbol{u}^{(k)} \circledast \boldsymbol{g}_k^{(j)} = \sum_{i=1}^{k} \boldsymbol{u}^{(i)} \circledast \boldsymbol{g}_i^{(j)}\ ,$$

where $\circledast$ is the convolution operator hence the name convolutional codes.

Convolutional encoders can be implemented by using shift registers with m memory elements. As an example Fig. 2.3 shows a rate $\frac{1}{2}$ convolutional code

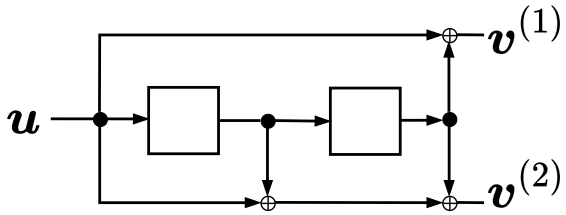


Figure 2.3: A feed forward encoder for a rate $\frac{1}{2}$ convolutional code.

with one input bit and two output bits with $m = 2$. The generator vectors are usually specified using the octal representation. That is the bits are divided into tuples of size 3 from the right and each tuple is converted to decimal with the leftmost bit as most significant. The generator vectors for the encoder with their octal representation are given as

$$\begin{aligned} \boldsymbol{g}_1^{(1)} &= \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} = (5) \\ \boldsymbol{g}_1^{(2)} &= \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} = (7)\,. \end{aligned}$$

The code can then be abbreviated as a (5,7) CC. The encoding of convolutional codes can also be represented by a state diagram with the state being determined by the contents of the shift registers. Fig. 2.4 shows the state diagram for the (5,7) encoder. The states are represented by the circles while each transition arrow is marked by the input/output bits. Expanding the state diagram for

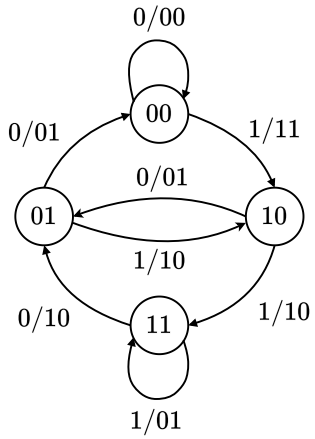


Figure 2.4: State diagram for the (5,7) CC encoder in Fig. 2.3.

consecutive time instants we obtain what is known as the trellis of the code. The trellis for the (5,7) code is shown in Fig. 2.5.

2.4.3 BCH Codes

Bose–Chaudhuri–Hocquenghem codes (BCH codes) form a class of cyclic error-correcting codes that are constructed using polynomials over a Galois field [29, 31]. The decoding of BCH codes is very efficient since it uses algebraic methods

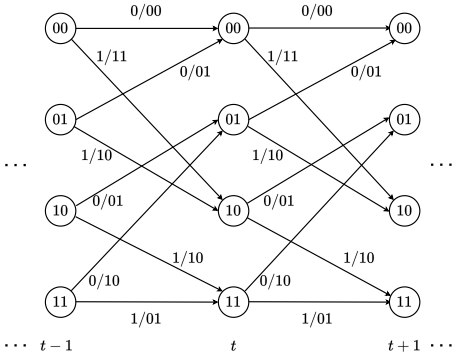


Figure 2.5: Trellis diagram for the (5,7) CC encoder in Fig. 2.3.

which can be efficiently implemented in hardware. An interesting feature of BCH codes is that one can design precisely a code which can correct t errors. That is for every integers $m \geq 3$ and $t < 2^{m-1}$, there exists a BCH code with the following parameters,

Block length:	$N = 2^m - 1$
Number of parity-check digits:	$N - K \leq mt$
Minimum distance:	$d_{\min} \geq 2t + 1$.

A binary $N-$tuple $(v_0, v_1, \cdots, v_{N-1})$ is a codeword of a t-error correcting BCH code if and only if the polynomial $\boldsymbol{v}(X) = v_0 + v_1 \cdot X + v_2 \cdot X^2 + \cdots + v_{N-1} \cdot X^{N-1}$ has $\alpha, \alpha^2, \ldots \alpha^{2t}$ as roots. This means for every codeword we have

$$\boldsymbol{v}(\alpha^i) = v_0 + v_1 \cdot \alpha^i + v_2 \cdot \alpha^{2i} + \cdots + v_{N-1}\alpha^{(N-1)i} = 0$$

for $1 \leq i \leq 2t$. In matrix notation this can be written as

$$(v_0, v_1, v_2 \cdots v_{N-1}) \cdot \begin{pmatrix} 1 \\ \alpha^i \\ \alpha^{2i} \\ \vdots \\ \alpha^{(N-1)i} \end{pmatrix} = 0 \, .$$

With this it can be seen that the parity-check matrix $\boldsymbol{H}$ is given as

$$\boldsymbol{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{N-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^2)^3 & \cdots & (\alpha^2)^{N-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{N-1} \\ \vdots & & & & & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & (\alpha^{2t})^3 & \cdots & (\alpha^{2t})^{N-1} \end{pmatrix} .$$

For an element of GF(2^m) represented as $a = \alpha^i$, every element represented as $(\alpha^i)^{2^l}$ is a conjugate of a. If α^j is a conjugate of α^i for some i and j then α^j is a root of a polynomial $\boldsymbol{v}(X)$ with coefficients in GF(2) if and only if α^i is a

root of $\boldsymbol{v}(X)$. We can thus omit the rows of $\boldsymbol{H}$ which correspond to conjugates of any other row. This implies that the matrix $\boldsymbol{H}$ can be reduced to the form

$$\boldsymbol{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{N-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{N-1} \\ \vdots & & & & & \vdots \\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & (\alpha^{2t-1})^3 & \cdots & (\alpha^{2t-1})^{N-1} \end{pmatrix}.$$

To get the binary representation of $\boldsymbol{H}$ we represent each element of $\mathrm{GF}(2^m)$ by its binary representation in column form.

Example 2.4.1. *Consider as 2-error correcting BCH code with $m = 3$. We have $N = 2^3 - 1 = 7$. The parity-check matrix is given as*

$$\boldsymbol{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \end{pmatrix} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \end{pmatrix}.$$

Here we have used the fact that $\alpha^7 = 1$. The corresponding binary representation is given as

$$\boldsymbol{H} = \left(\begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right).$$

2.5 Decoders with Soft Information

The bit-wise MAP decoder is the optimal decoder if we aim to minimize the bit error rate (BER). For a received sequence $\boldsymbol{y}$ the bit-wise MAP decoder is given as

$$\hat{u}_i = \arg \max_{u_i \in \{0,1\}} P(u_i|\boldsymbol{y}).$$

Soft decoders output the *a posterior* probability (APP) in terms of log-likelihood ratio instead of the hard decision. The log-likelihood ratio $\mathrm{L}(u_i)$ is commonly called the L-value and is given by

$$\mathrm{L}(u_i) = \ln \frac{P(u_i = 0|\boldsymbol{y})}{P(u_i = 1|\boldsymbol{y})}.$$

Where $\ln(x)$ is the natural logarithm of x.

The soft information makes the decoders more effective when they are used in iterative receivers.

Chapter 3

Graph Based Systems

Computations involving functions with many variables can be very complex. If the function can be factorized into smaller functions, each with a small set of the variables, the computation complexity can be reduced significantly by exploiting this factorization [32]. This factorization can be modelled by factor graphs. This chapter introduces factor graphs and briefly discusses some applications in communication systems, particularly codes on graphs and iterative receivers for channels with ISI.

3.1 Factor Graphs

A factor graph is bipartite graph with one set of nodes consisting of constraints and the other set consisting of variables. The constraints represent conditions which the variables connected to the constraint node must fulfill. In other words, these are functions on a set of variables nodes.

Example 3.1.1. *Consider a function $f(x_1, x_2, x_3, x_4, x_5)$ with five variables. If the function can be expressed as a product*

$$f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_2) f_2(x_1, x_3, x_4) f_3(x_3, x_5) f_4(x_4, x_6)$$

with four factors, the factor graph for the function can be drawn as shown in Fig. 3.1.

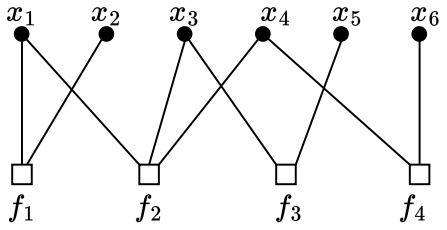


Figure 3.1: (a) An example of a factor graph with constraints as squares and variables as circles.

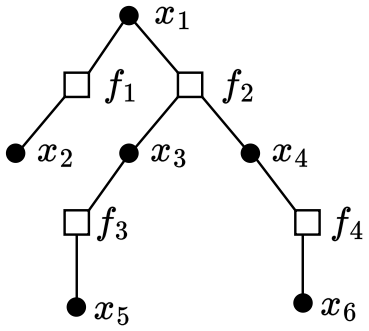


Figure 3.2: (a) The factor graph in Fig. 3.1 redrawn with the root at x_1.

3.1.1 Message Passing on Graphs

It is a typical problem to compute marginals for given a function. For example we might be interested in computing the marginal $f(x_1)$ for the function $f(x_1, x_2, x_3, x_4, x_5)$ in Example 3.1.1. To do this we have to sum over all variables except x_1. Using the notation $x_{\sim i}$ to denote all variables except x_i we have

$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5) =$$
$$\left[\sum_{x_2} f_1(x_1, x_2)\right]\left[\sum_{x_3, x_4} f_2(x_1, x_3, x_4) \sum_{x_5} f_3(x_3, x_5) \sum_{x_6} f_4(x_4, x_6)\right].$$

This marginalization can be modelled by a tree rooted at variable x_1 as shown in Fig. 3.2. We can thus compute the marginalization by passing messages in the graph starting with leaf nodes. Each node passes a message to its parent node by aggregating messages from its children according to some predefined rule. We need a tree rooted at x_i for each variable x_i we want to marginalize. We could however achieve marginalization with respect to all variables by letting each node pass a message to each edge connected to it based incoming messages from other edges. The message passed to the node connected to the edge is *extrinsic* to the node in the sense that it does not derive any information from the destination node. If the graph has no cycles this message passing will converge to the optimal global solution after many iterations. The presence of cycles violates the extrinsic assumption thus leading to degrading of performance. This message passing on graphs is also called *belief propagation* (BP) when the messages passed are probabilities.

3.2 Codes on Graphs

Codes can be modelled on bipartite (Tanner) graphs. This idea was introduced by Tanner [33] and is extensively used to model how larger codes can be constructed from smaller codes. In this section we discuss examples of codes on graphs of interest in this thesis staring with LDPC codes and finishing with serially concatenated turbo like codes.

3.2.1 Low-Density Parity-Check Codes

Low-density parity-check (LDPC) codes is a class of linear codes characterized by having a parity-check matrix $\boldsymbol{H}$, with a very small fraction of ones. The parity-check matrix can be represented by a Tanner graph with the bits as the variable nodes and the checks as the constraint nodes. The parity-check matrix is actually the adjacency matrix of the graph in the sense that an entry (i, j) of $\boldsymbol{H}$ is 1 if there is an edge connecting CN i to VN j otherwise the entry is zero. This can be seen as a concatenation of repetition code (the variables) and single parity-check code (the checks). These codes, first introduced by Gallager [5], can be decoded with low complexity by message passing on a graph.

Example 3.2.1. *Consider a linear code with parity-check matrix*

$$\boldsymbol{H} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}, \tag{3.1}$$

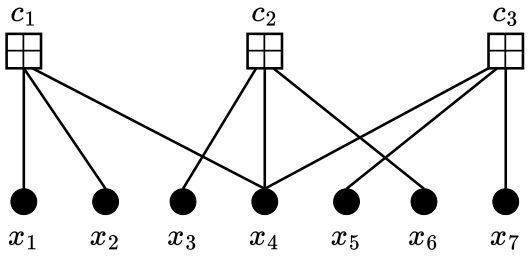
where x_j is an element of binary Galois field ($\mathbb{F}_2$). For any $\boldsymbol{x} \in \mathbb{F}_2^7$ we have the membership test function define as

$$f(x_1, \dots x_7) = \begin{cases} 1, & \text{if } \boldsymbol{H}\boldsymbol{x}^{\mathsf{T}} = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{3.2}$$

In other words $f(x_1, \dots x_7)$ is an indicator function whether the sequence is a valid codeword of the code defined by the parity-check matrix $\boldsymbol{H}$. This can be factored by considering each check equation separately as

$$f(x_1, \dots x_7) = \mathbb{I}\{x_1 + x_2 + x_4\} \cdot \mathbb{I}\{x_3 + x_4 + x_6\} \cdot \mathbb{I}\{x_4 + x_5 + x_7\}.$$

The corresponding Tanner factor graph is shown in the figure below. The constraint nodes in this graph are the check nodes which are represented using squares with a plus sign while the VNs are represented by dark circles.



LDPC codes can either be regular or irregular. With regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ codes, where d_{v} and d_{c} are the VN and CN degrees respectively, each variable node is connected to d_{v} edges while each check node has d_{c} edges. For irregular codes on the other hand the nodes have a range of degrees. The degree distribution can be characterized either from the node or edge perspective using polynomials.

We describe the degree distribution following the notation in [34]. For the node perspective we have

$$\mathsf{L}(x) = \sum_i \mathsf{L}_i x^i, \qquad \mathsf{R}(x) = \sum_i \mathsf{R}_i x^i \,.$$

Here L and R are fractions of VNs and CNs respectively with degree i.

In many scenarios it is more convenient to express the degree distribution from the edge perspective. That is

$$\lambda(x) = \sum_i \lambda_i x^{i-1}, \qquad \rho(x) = \sum_i \rho_i x^{i-1} \,,$$

with λ_i and ρ_i as the fraction of edges which are connected to VNs and CNs respectively with degree i. The two perspectives are related as

$$\lambda(x) = \frac{\mathsf{L}'(x)}{\mathsf{L}'(1)}$$
$$\rho(x) = \frac{\mathsf{R}'(x)}{\mathsf{R}'(1)} \,.$$

We use $f'(x)$ to denote the derivative of $f(x)$.

The design rate of a code is defined as the rate of the code if all check equations are linearly independent. This is given as

$$R(\lambda, \rho) = 1 - \frac{\mathsf{L}'(1)}{\mathsf{R}'(1)} = 1 - \frac{\int_0^1 \rho(x)}{\int_0^1 \lambda(x)} \,,$$

where $\mathsf{L}'(1)$ and $\mathsf{R}'(1)$ represents the average degree of the VNs and CNs respectively. For example for a regular $(d_\mathsf{v}, d_\mathsf{c})$ LDPC code we have

$$R = 1 - \frac{d_\mathsf{v}}{d_\mathsf{c}} \,.$$

SISO Decoders for LDPC Codes

The outgoing extrinsic L-values form a VN to an edge $e_{k',j}$ is given as the sum of the L-values from the channel L_ch and the incoming edges from the other edges. That is

$$\mathrm{L}_\mathsf{v}^{(\ell)}\left(e_{k,j}\right) = \mathrm{L}_\mathrm{ch}(j) + \sum_{k' \neq k} \mathrm{L}_\mathsf{c}^{(\ell-1)}\left(e_{k',j}\right) \,. \tag{3.3}$$

This is illustrated in Fig. 3.3 (a). At the CN, Fig. 3.3(b), the update rule is given as

$$\mathrm{L}_\mathsf{c}^{(\ell)}\left(e_{k,j}\right) = \boxplus_{j' \neq j} \mathrm{L}_\mathsf{v}^{(\ell-1)}\left(e_{k,j'}\right) \,, \tag{3.4}$$

where the box plus operator $\boxplus$ is defined as

$$\boxplus_{j'} \mathrm{L}_\mathsf{v}^{(\ell-1)}\left(e_{k,j'}\right) = 2 \tanh^{-1}\left(\prod_{j'} \tanh\left(\frac{1}{2}\mathrm{L}_\mathsf{v}^{(\ell-1)}\left(e_{k,j'}\right)\right)\right) . \tag{3.5}$$

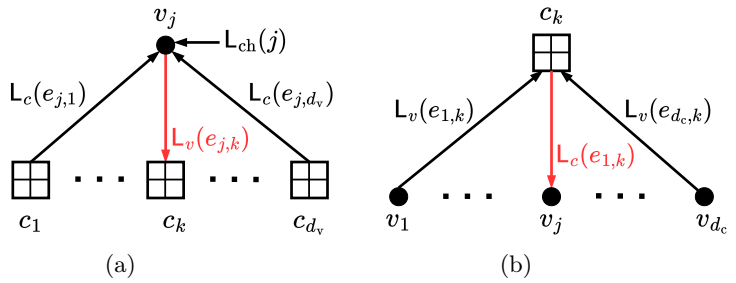


Figure 3.3: Message passed from (a) CN to VN (b) VN to CN. The edges with outgoing message are in red color while those with incoming messages are in black.

The operation can be done by taking two values at a time. For pair of L-values we have

$$L(v_1) \boxplus L(v_2) = \ln \frac{1 + e^{L(v_1)} e^{L(v_2)}}{e^{L(v_1)} + e^{L(v_2)}}$$

LDPC Codes Constructed from Protographs

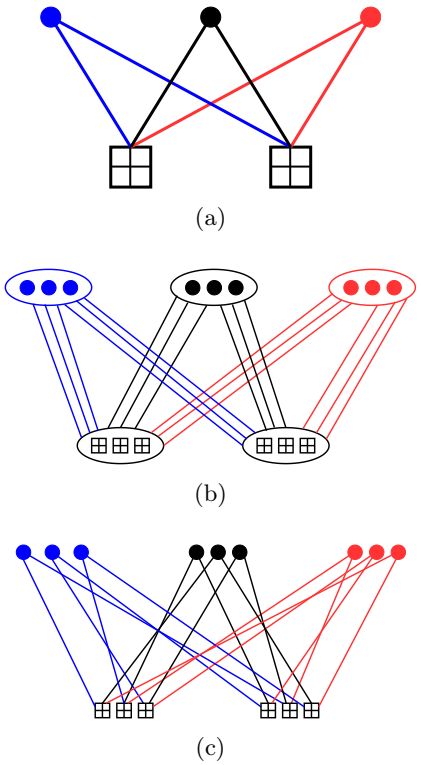
It is becoming common to construct LDPC codes with some predefined structure. A relatively small base graph called the protograph is first designed and larger graphs of various sizes can be constructed by copying the base graph followed by permutation, a process called lifting. The process preserves the rate, the degree distribution and the computation graph. This can significantly reduce the decoding complexity compared to randomized constructions and has been adopted in various standards such as 5G long term evolution (LTE), wireless local area network (WLAN), DVB-S2 and worldwide interoperability for microwave access (WiMAx). As an example consider the protograph shown in Fig. 3.4(a). The protograph can be represented as

$$\boldsymbol{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The protograph has three VN types and two CN types whereby each VN has degree 2 and every CN has degree 3. To lift it by a factor of three, each node and edge is copied three times as shown in (b). We then apply a random permutation of the edges to obtain the full graph in (c).

3.2.2 Convolutional Codes: Graphical Representation

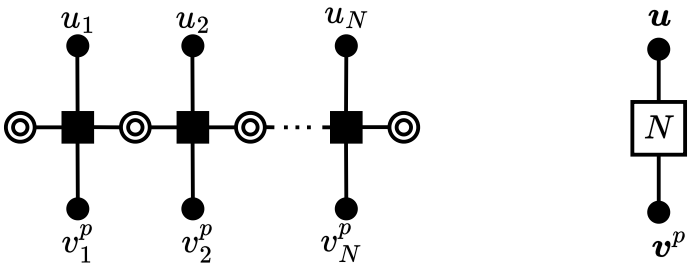
Convolutional codes can also be represented by factor graphs. To do this we have to introduce hidden VNs which represent the state of the encoder at each time instant. As an example Fig. 3.5(a) shows a factor graph for a rate 1/2 systematic convolutional code. The trellis section has length N. The dark circles represent the VN corresponding to code bits while the double circles represent the VN corresponding to the state nodes. The black squares correspond to the constraints governing the state transition and the input output bits for the encoder. For convenience we represent the factor graph using compact



(a) (b) (c)

Figure 3.4: Protograph with 3 VN types and 2 CN types. (b) Copying the protograph 3 times (c) Permuting the edges.

graph notation. This is achieved by grouping nodes with similar distributions into a single node. Fig. 3.5(b) shows the compact representation of the same factor graph. We have the systematic bits node $\boldsymbol{u}$ and the parity node $\boldsymbol{v}$. The constraints and state node are lumped into a square showing the length of the trellis section.



(a) (b)

Figure 3.5: (a) Factor graph representation of a rate 1/2 systematic CC. (b) Compact graph representation of the CC.

3.2.3 The BCJR Algorithm

The BCJR algorithm, named after its invetors Bahl, Cocke, Jelinek and Raviv [35] is used for APP decoding for systems defined on trellises. It is widely used for soft decoding of convolutional codes as well as equalization of ISI channels. We briefly present the algorithm for a systematic convolutional code with input bits $\boldsymbol{u} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N)$ and output bits $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N)$. It is important to note that for a systematic CC $\boldsymbol{v}$ contains both the information bits $\boldsymbol{u}$ and the parity bits $\boldsymbol{v}^p$. The transmitted symbols are assumed to be modulated by BPSK to symbol sequence $\boldsymbol{x} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. The received sequence is denoted as $\boldsymbol{y}$.

Consider a trellis section between the time instants $t-1$ to t with the state at time $t-1$ denoted as σ' and the state at time t denoted by σ. The joint probability of the state transition from σ' to σ and the received sequence $\boldsymbol{y}$ can be factored as

$$\mathrm{P}_{S_{t-1},S_t,\boldsymbol{Y}_t}\big(\sigma',\sigma,\boldsymbol{y}_t\big) = \mathrm{P}_{S_{t-1},\boldsymbol{Y}_{<t}}\big(\sigma',\boldsymbol{y}_{<t}\big) \cdot \mathrm{P}_{S_t,\boldsymbol{Y}_t|S_{t-1}}\big(\sigma,\boldsymbol{y}_t|\sigma'\big) \cdot \mathrm{P}_{\boldsymbol{Y}_{>t}|S_t}\big(\boldsymbol{y}_{>t}|\sigma\big)\,.$$

Defining

$$\begin{aligned}
\alpha_{t-1}(\sigma') &= \mathrm{P}_{S_{t-1},\boldsymbol{Y}_{<t}}\big(\sigma',\boldsymbol{y}_{<t}\big)\\
\beta_t(\sigma) &= \mathrm{P}_{\boldsymbol{Y}_{>t}|S_t}\big(\boldsymbol{y}_{>t}|\sigma\big)\\
\gamma_t(\sigma',\sigma) &= \mathrm{P}_{S_t,\boldsymbol{Y}_t|S_{t-1}}\big(\sigma,\boldsymbol{y}_t|\sigma'\big)\,,
\end{aligned}$$

we can rewrite the factorization as

$$\mathrm{P}_{S_{t-1},S_t,\boldsymbol{Y}_t}\big(\sigma',\sigma,\boldsymbol{y}_t\big) = \alpha_{t-1}(\sigma') \cdot \gamma_t(\sigma',\sigma) \cdot \beta_t(\sigma)\,.$$

The branch metric is computed based on the observed symbols $\boldsymbol{y}_t$ and the a priori probabilities of the input bits $\mathrm{P}\big(\boldsymbol{u}_t\big)$. That is

$$\gamma_t(\sigma',\sigma) = \mathrm{P}\big(\boldsymbol{y}_t|\boldsymbol{x}_t\big) \cdot \mathrm{P}\big(\boldsymbol{u}_t\big)\,.$$

For different scenarios or systems, the branch metric computation might differ. That is it depends on the channel model, the target bits and the type of code concatenation. The values of α_t and β_t are computed recursively in what is known as the forward and backward recursion respectively. That is

$$\begin{aligned}
\alpha_t(\sigma) &= \sum_{\sigma'} \alpha_{t-1}(\sigma') \cdot \gamma_t(\sigma',\sigma) \quad \text{for } t = 1,\ldots N-1\,,\\
\beta_{t-1}(\sigma') &= \sum_{\sigma} \beta_t(\sigma) \cdot \gamma_t(\sigma',\sigma) \qquad \text{for } t = N,\ldots 2\,.
\end{aligned}$$

If the initial state is known (e.g. the zero state), the boundary condition for $\alpha_t(\sigma)$ is set as

$$\alpha_0(\sigma) = \begin{cases} 1 & \text{if } \ \sigma = 0\,,\\ 0 & \text{otherwise.} \end{cases}$$

The same condition applies for $\beta_{N+1}(\sigma)$ if the encoder is terminated to end in the zero state. Otherwise all states are given equal probability which for convenience can be set to 1.

The L-value for a given bit u_i which is part of the input at time t is computed as

$$\mathrm{L}(u_i) = \ln \frac{\sum_{(\sigma',\sigma):u_i=0} = \alpha_{t-1}(\sigma') \cdot \gamma_t(\sigma',\sigma) \cdot \beta_t(\sigma)}{\sum_{(\sigma',\sigma):u_i=1} = \alpha_{t-1}(\sigma') \cdot \gamma_t(\sigma',\sigma) \cdot \beta_t(\sigma)} .$$

The prior knowledge about u_i is contained in this L-value. Most often, the extrinsic L-value $\mathrm{L}^e(u_i)$ is needed which does not contain the prior of u_i. This is achieved by subtracting the prior L-value $\mathrm{L}^a(u_i)$ from $\mathrm{L}(u_i)$.

3.2.4 Turbo-like Codes

Turbo codes were invented in the early 90s by Berrou *et al.* [3]. The codes consisted of parallel concatenated convolutional codes which are decoded by exchange of information between the two decoders. This technique provided the first practical codes with performance close to the Shannon limit. We highlight the parallel concatenated codes (PCC) and the serially concatenated codes (SCC).

Parallel Concatenated Codes

The idea behind PCC or turbo codes is to encode a sequence $\boldsymbol{u}$ using two (or more) encoders. An example with two systematic convolutional codes each with rate 1/2 together with the corresponding factor graph is shown in Fig. 3.6. The

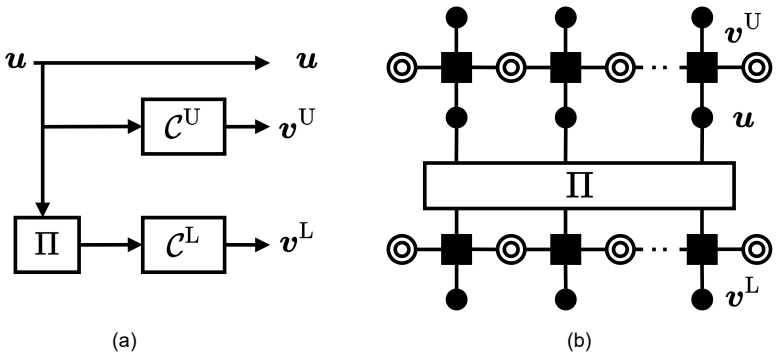


Figure 3.6: A turbo code with two systematic convolutional encoders. The encoder is shown in (a) while the factor graph is shown in (b).

sequence $\boldsymbol{u}$ is encoded directly by the upper encoder $\mathcal{C}^{\mathrm{U}}$ to produce parity bits $\boldsymbol{v}^{\mathrm{U}}$. The lower encoder $\mathcal{C}^{\mathrm{L}}$ receives a permuted version of $\boldsymbol{u}$ and produces the parity bits $\boldsymbol{v}^{\mathrm{L}}$. The output of the encoder is then $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$ making the overall rate be 1/3. Different rates can be achieved by puncturing.

Serially Concatenated Turbo Codes

Another variation of turbo codes consists of serially concatenated codes. The information sequence $\boldsymbol{u}$ is encoded by the outer encoder $\mathcal{C}^{\mathrm{O}}$ producing the parity bits $\boldsymbol{v}^{\mathrm{O}}$. The sequence $(\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}})$, after reordering by an interleaver, is used as the input of another encoder, the inner encoder $\mathcal{C}^{\mathrm{I}}$ to produce parity bits $\boldsymbol{v}^{\mathrm{I}}$. The final output of the encoder is then $(\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}}, \boldsymbol{v}^{\mathrm{I}})$. An example is shown in Fig. 3.7

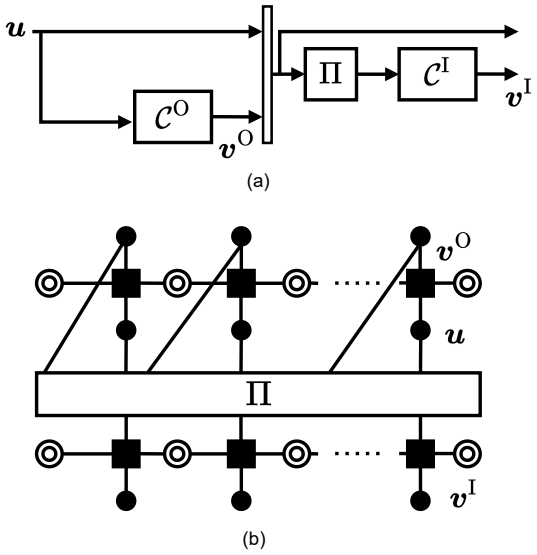


Figure 3.7: A SCC with two systematic convolutional encoders. The encoder is shown in (a) while the factor graph is shown in (b).

where the component codes are both systematic, rate 1/2 convolutional codes. The resulting code rate is 1/4. The BCJR algorithm for the outer code has to be modified when computing the branch metric by incorporating the prior form the inner decoder for the systematic and parity bits.

3.3 EXIT Charts

Extrinsic information transfer (EXIT) charts provide a graphical way to visualize the convergence of iterative systems assuming infinitely long blocks [36]. The parameter tracked is the mutual information between the messages passed between the components and the corresponding code bits. That is with code bits V and messages L we have the mutual information as

$$\begin{aligned} I(V;\mathrm{L}) &= H(V) - H(V|\mathrm{L}) \\ &= 1 - H(V|\mathrm{L})\,, \end{aligned}$$

assuming the bits are equally likely to be 0 or 1. In some context it is common to use the conditional entropy $H(V|\mathrm{L})$ instead of the mutual information. For a given decoder, we have the mutual information between V and the a priori messages at the input $I_a(V;\mathrm{L})$ and the mutual information between V and the extrinsic messages $I_e(V;\mathrm{L})$.

3.3.1 EXIT Charts on the BEC

For the BEC the L-values are either $\pm\infty$ if the bits are known and 0 if they are erasures. Let p be the probability that a bit is still an erasure after some decoding iterations. With known bits the relative entropy $H(V|\mathrm{L}) = 0$ with probability $1-p$ and with erasures $H(V|\mathrm{L}) = H(V) = 1$ with probability p.

The mutual information is then given as $I(V;\mathrm{L}) = 1-p$. For the BEC we have an exact transfer function which can be deduced from the erasure probability.

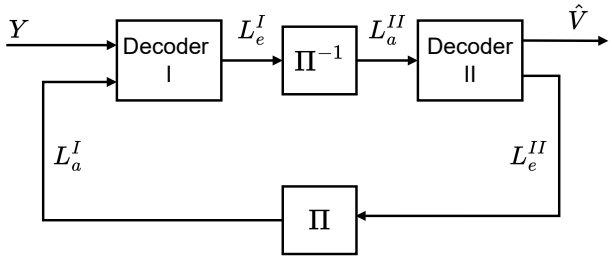


Figure 3.8: Block diagram showing the exchange of soft L-values between two decoders. Decoder I has in this case access to the received symbols Y while Decoder II makes the final estimate of the code bits $\hat{V}$.

The mutual information at the output is a function of the mutual information at the input (and the channel condition if the decoder has access to the received symbols). That is

$$I_e(V;\mathrm{L}) = f\left(I_a(V;\mathrm{L})\right) .$$

If we have two decoders, Decoder I and II as shown in Fig. 3.8 , the extrinsic information at the output of Decoder I, I_e^{I}, becomes the a priori information at Decoder II, I_a^{II}, and vice versa. When plotting the transfer functions, the other function say f_{II} is plotted as an inverse. For the system to converge the two curves should not intersect at any other point than the point $(1,1)$, the point where at least one decoder has no uncertainty about the code bits.

For example for an LDPC code we have the check transfer function given as $g(x) = \rho(x)$ and the variable transfer function given as $f(x;\varepsilon) = 1-\varepsilon\lambda(1-x)$ where ε is the channel erasure probability. For example for a regular $(3,6)$ LDPC code we have

$$\begin{aligned} g(x) &= x^5 \\ f(x;\varepsilon) &= 1-\varepsilon(1-x)^2 , \end{aligned}$$

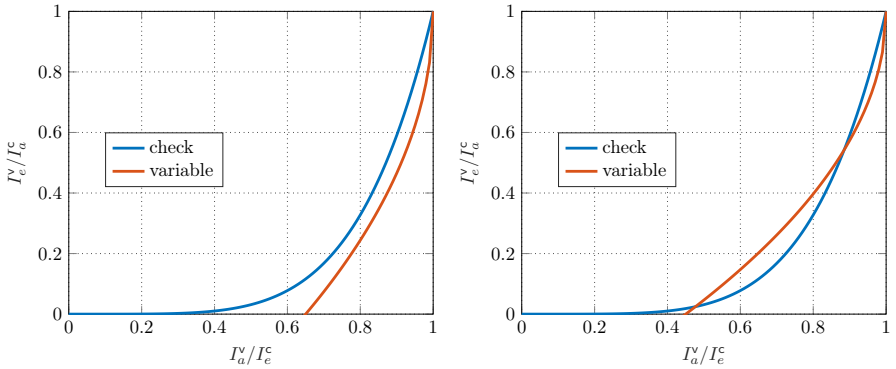
which implies $f^{-1}(x;\varepsilon) = 1-\sqrt{\frac{1-x}{\varepsilon}}$. Fig. 3.9 shows the EXIT chart for the code with channel erasure probabilities $\varepsilon = 0.35$ and $\varepsilon = 0.55$. The BP threshold $\varepsilon_{\mathrm{BP}} \approx 0.429$. It is clear from the chart that with $\varepsilon > \varepsilon_{\mathrm{BP}}$ the trajectory would not converge since the two curves cross each other.

3.3.2 Generalized EXIT Function and MAP Threshold

Consider the sequence of symbols $\mathcal{X} = (X_1, \ldots, X_n)$ transmitted through a BEC with erasure probability ε for all symbols and received symbols $\mathcal{Y} = (Y_1, \ldots, Y_n)$. The Generalized EXIT (GEXIT) function is defined as [37]

$$G(\varepsilon) = \lim_{n\to\infty} \frac{1}{n} \frac{\partial H(\mathcal{X}|\mathcal{Y}(\varepsilon))}{\partial \varepsilon} = \frac{1}{n} \sum_{i=1}^{n} H(X_i|\mathcal{Y}(\varepsilon)) .$$

This represents the average conditional entropy for all symbols for a given channel parameter ε assuming a MAP receiver. The area below the GEXIT



(a) $\varepsilon < \varepsilon_{\mathrm{BP}}$; No overlap (b) $\varepsilon > \varepsilon_{\mathrm{BP}}$; Overlap

Figure 3.9: EXIT chart of a $(3, 6)$ LDPC code for (a) $\varepsilon = 0.35$ (b) $\varepsilon = 0.55$.

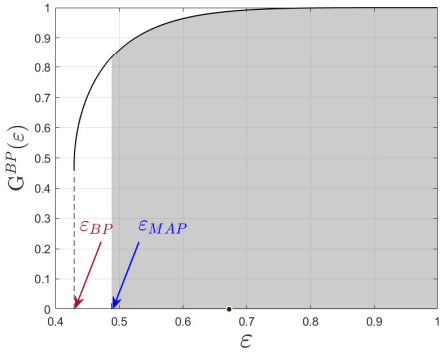


Figure 3.10: GEXIT for a (3,6) LDPC code showing the upper bound on the MAP threshold. In this case $\varepsilon^{\mathrm{MAP}} \approx 0.4881$.

curve equals the rate of the code. This phenomenon is known as the area theorem [37]. The BP GEXIT function, $G^{\mathrm{BP}}(\varepsilon)$, represents the conditional entropy when the BP receiver is used instead (with infinite number of iterations). Since the MAP receiver is better than the BP receiver we have

$$G^{\mathrm{BP}}(\varepsilon) \geq G(\varepsilon)\,.$$

This together with the area theorem for GEXIT function can be used to derive an upper bound on the MAP threshold from the BP GEXIT curve. That is for a code with rate R the MAP threshold $\varepsilon^{\mathrm{MAP}}$ is upper bounded by a positive number $\bar{\varepsilon}$ such that

$$\int_{\bar{\varepsilon}}^{1} G^{\mathrm{BP}}(\varepsilon) d\varepsilon = R\,.$$

Fig. 3.10 shows an example for the case of a regular $(3, 6)$ LDPC code. In the figure, the shaded area for $0.4881 \leq \varepsilon \leq 1$ equals $\frac{1}{2}$ which is the rate of the code.

3.3.3 EXIT Charts for the AWGN Channel

For a channel with AWGN the L-values can take any real values in $[-\infty, +\infty]$. The mutual information is then computed on the distribution of the L-values. That is

$$\begin{aligned} I(V;\mathrm{L}) &= \sum_{x=-1,+1} \mathrm{P}_X(x) \int_{-\infty}^{+\infty} \mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x) \ln \frac{\mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x)}{\sum_{x=-1,+1} \mathrm{P}_X(x)\mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x)} d\mathrm{L} \\ &= \frac{1}{2} \sum_{x=-1,+1} \int_{-\infty}^{+\infty} \mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x) \ln \frac{2\mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x)}{\mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|+1) + \mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|-1)} d\mathrm{L}\,. \end{aligned}$$

The second equality follows from the equal likely assumption i.e. $\mathrm{P}_X(x) = 1/2$. It is common to use the assumption that the distribution $\mathrm{f}_{\mathrm{L}|X}(\mathrm{L}|x)$ is Gaussian with variance σ_L^2 and mean $\frac{\sigma_L^2}{2}$. The mutual information is computed as a function of σ_L using what is called the J-function. That is

$$I(V;\mathrm{L}) = J(\sigma_L) = 1 - \int_{-\infty}^{+\infty} \frac{\exp\left(-\frac{(\mathrm{L}-\sigma_L^2/2)}{2\sigma_L^2}\right)}{\sigma_L\sqrt{2\pi}} \ln\Big(1+\exp(-\mathrm{L})\Big) d\mathrm{L}\,.$$

The inverse of the J-function is computed using some numerical approximations or using a look-up table. For the BI-AWGN channel with noise variance σ_w^2, σ_L is given as

$$\sigma_L = \frac{2}{\sigma_w}\,.$$

For a channel with AWGN, the behaviour of the iterative systems can be predicted from the EXIT chart with very limited accuracy. This is because, by passing only one parameter of the distribution between the components in the iterative receiver, namely the mutual information, we lose much information about the actual distribution. A more accurate analysis would require passing the actual distributions between the component decoders. This is what is done in density evolution.

3.4 Density Evolution

The soft messages (the L-values) passed between components in an iterative system has a distribution which evolves with iterations until it converges to some stable distribution depending on the channel quality. Tracking the density of these messages is what is known as density evolution (DE). The density may converge to the zero error state or not.

3.4.1 Density Evolution on the BEC

Since the BEC cannot introduce errors, the L-values passed between the CNs and VNs can either be of infinite magnitude (when the symbol is known) or zero (when the symbol is unknown). Assuming the all-zero codeword we can simplify the analysis without losing information about the density of the L-values by just tracking the probability that the message is an erasure. Here

we present a DE for a randomized construction of LDPC codes where we could consider a single edge type.

Let $p^{(\ell)}$ and $q^{(\ell)}$ be the probability that a message from a CN and VN respectively, to an edge is an erasure. Consider a VN with degree d_{v}. The message from a VN to an edge is an erasure if the symbol was erased by the channel and all incoming messages from the $d_{\mathsf{v}} - 1$ edges are erasures. That is,

$$p^{(\ell)} = \varepsilon \left(q^{(\ell-1)} \right)^{d_{\mathsf{v}}-1} .$$

Hereby the message passed from CN with degree d_{c} to an edge is an erasure if any of the incoming messages from the other $d_{\mathsf{c}} - 1$ edges is an erasure. That is,

$$q^{(\ell)} = 1 - \left(1 - p^{(\ell-1)} \right)^{d_{\mathsf{c}}-1} .$$

For a general LDPC code we have the DE equations given as

$$\begin{aligned} p^{(\ell)} &= \varepsilon \lambda \left(q^{(\ell-1)} \right) \\ q^{(\ell)} &= 1 - \rho \left(1 - p^{(\ell-1)} \right) \end{aligned}$$

with the initialization $q^{(0)} = 1$.

3.4.2 Density Evolution on the AWGN Channel

For a BI-AWGN channel without memory, the L-values can take a range of real values from $-\infty$ to ∞. We cannot have the density evolution equations in a closed form. We could however use a discretized DE as described in [7]. At the VN side with the update rule given in Section 3.3, the density of the outgoing message from a VN v to an edge $e_{k',j}$, $\mathsf{p}\left(\mathrm{L}_{\mathsf{v}}^{(\ell)}\left(e_{k,j}\right)\right)$, is given as the convolution of the density of the channel L-value L_{ch} and $d_{\mathsf{v}} - 1$ densities of incoming messages form the other edges. In case of protograph based LDPC codes each edge will have its own density $\mathsf{p}\left(\mathrm{L}_{\mathsf{c}}^{(\ell-1)}\left(e_{k,j}\right)\right)$. For the unstructured LDPC codes, however, we have only a single average density $\mathsf{p}\left(\mathrm{L}_{\mathsf{c}}^{(\ell-1)}\right)$ from CN and the density of the extrinsic L-values is given by

$$\mathsf{p}\left(\mathrm{L}_{\mathsf{v}}^{(\ell)}\right) = \mathsf{p}\left(\mathrm{L}_{\mathrm{ch}}\right) \circledast \mathsf{p}\left(\mathrm{L}_{\mathsf{c}}^{(\ell-1)}\right)^{\circledast(d_{\mathsf{v}}-1)} ,$$

where $\circledast$ represents convolution of densities. For an arbitrary VN distribution we have the output density

$$\mathsf{p}\left(\mathrm{L}_{\mathsf{v}}^{(\ell)}\right) = \mathsf{p}\left(\mathrm{L}_{\mathrm{ch}}\right) \circledast \sum_i \lambda_i \mathsf{p}\left(\mathrm{L}_{\mathsf{c}}^{(\ell-1)}\right)^{\circledast(i-1)} .$$

Similarly for the extrinsic L-values from the check node for single edge type LDPC codes we have

$$\mathsf{p}\left(\mathrm{L}_{\mathsf{c}}^{(\ell)}\right) = \sum_i \rho_i \mathsf{p}\left(\mathrm{L}_{\mathsf{v}}^{(\ell-1)}\right)^{\boxed{*}(i-1)} .$$

The symbol $\boxed{*}$ represents the density transformation at the CN according to the CN update rule. To speed up the computation the convolution at the VN is implemented using an FFT and a lookup table is used for the density transformation at the CN.

3.5 Channels with Intersymbol Interference

In this section we discuss channels with intersymbol interference (ISI). We first briefly discuss the underlying continuous model and how the discrete model is derived from it. We also introduce turbo equalization and capacity limits for ISI channels.

3.5.1 Discrete Model from Continuous Model

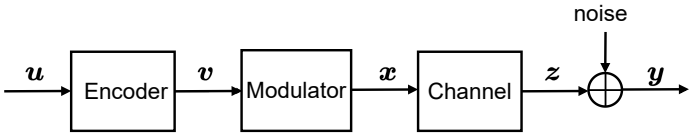


Figure 3.11: Block diagram showing the transmitter and the ISI channel.

Consider a model shown in Fig 3.11. The transmitter accepts a block of K information bits, $\boldsymbol{u}$. The information bits are encoded by the channel encoder to produce N coded bits $\boldsymbol{v}$ which are put to the modulator which maps them to a block $\boldsymbol{x}$ of symbols according to some modulation scheme. Each symbol x_i contains q coded bits where q is the modulation order. We assume a pulse modulator whereby each symbol x_i is associated with a pulse $p(t)$. The symbols are transmitted after every symbol time T_s. We thus have the transmitted symbol $s(t)$ as

$$s(t) = \sum_i x_i p(t - iT_s)\,.$$

The received signal $r(t)$ is the convolution of $s(t)$ and the channel impulse response $h(t)$ corrupted by noise $w_c(t)$. The noise is in most parts assumed to be additive white Gaussian noise (AWGN). We thus have

$$\begin{aligned} y(t) &= h(t) * s(t) + w_c(t) \\ &= \sum_i x_i h(t) * p(t - iT_s) \\ &= \sum_i x_i c(t - iTs)\,. \end{aligned}$$

Where $c(t) = h(t) * p(t)$ is the combined impulse response of the pulse and channel. For an ideal channel $h(t)$ is just a scaled Dirac delta function $\alpha\delta(t)$. At the receiver we assume a matched filter matched to $c(t)$. We thus have the discrete output of the matched filter given as

$$y_i = \int_{-\infty}^{\infty} y(t) * c^*(t - iTs)dt\,.$$

This can be expressed as

$$y_i = \sum_{j=-L}^{L} g_j x_{i-j} + \eta_i \,, \tag{3.6}$$

where $g_j = \int_{-\infty}^{\infty} c(t)c^*(t - jTs)dt$, L is the memory of the channel and η_k is Gaussian colored noise with $E[\eta_{k+j}\eta_k^*] = g_j N_0$. Equation (3.6) represents what is called the Ungerboeck model [38]. In what is called the Forney model a whitening filter is applied to obtain white Gaussian noise w_k [39]. The Forney model can thus be expressed as

$$y_k = \sum_{j=0}^{L} h_j x_{k-j} + w_k \,, \tag{3.7}$$

where $\boldsymbol{h}$ is a sequence such that $\boldsymbol{g}$ is its auto correlation sequence [40]. In this thesis the Forney model is assumed for the general ISI problem unless stated otherwise.

3.5.2 Turbo Equalization for ISI Channels

To mitigate the effect of ISI an equalizer which we refer to as the detector is usually employed. Operating the detector and the decoder for the code separately does not result in good performance. An optimal receiver would be a joint MAP receiver taking into account both the code and channel constraint. The complexity of such a receiver is however prohibitive even for channels with short memories. To get reasonably good performance with feasible complexity, the detector and the decoder have to exchange information in a number of iterations in what is called turbo equalization. In this thesis we assume that the channel is known perfectly at the receiver. In some works, however, the channel estimation can also be included in the equalization loop.

Equalization with a MAP Detector

We consider a bit-wise MAP detector which selects the output bit which has the highest a posterior probability based on the received received block $\boldsymbol{y}$. That is,

$$\begin{aligned}
\hat{v}_i &= \underset{\mathsf{v}\in\{0,1\}}{\operatorname{argmax}} \; P(v_i = \mathsf{v}|\boldsymbol{y}) \,, \\
&= \underset{\mathsf{v}\in\{0,1\}}{\operatorname{argmax}} \sum_{\boldsymbol{v}:v_i=\mathsf{v}} P(\boldsymbol{v}|\boldsymbol{y}) = \underset{\mathsf{v}\in\{0,1\}}{\operatorname{argmax}} \sum_{\boldsymbol{v}:v_i=\mathsf{v}} \frac{P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v})}{P(\boldsymbol{y})} \,, \\
&= \underset{\mathsf{v}\in\{0,1\}}{\operatorname{argmax}} \sum_{\boldsymbol{v}:v_i=\mathsf{v}} P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v}) \,.
\end{aligned} \tag{3.8}$$

Assuming that the entries of $\boldsymbol{v}$ are uncorrelated, $P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v})$ can be factorized as

$$P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v}) = \prod_{j=1}^{N} P(y_j|v_j)P(v_j) \,.$$

The SISO detector computes the actual probabilities $P(v_i = 0|\boldsymbol{y})$ and $P(v_i = 1|\boldsymbol{y})$ and output the logarithm of their ratios instead of a hard decision $\hat{v}_i$. The output is called log-likelihood ratio and is given by

$$L(v_i) = \ln \frac{P(v_i = 0|\boldsymbol{y})}{P(v_i = 1|\boldsymbol{y})} .$$

We then use the BCJR algorithm [35] to compute these log-likelihood ratios. The branch metric from state σ' to σ for the BCJR algorithm for equalization with bit-wise MAP detector is given as

$$\gamma(\sigma', \sigma) = P(y_j|v_j)P(v_j) = \exp\left(-\frac{|y_j - v_{\sigma',\sigma})|^2}{N_0}\right) ,$$

where $v_{\sigma',\sigma}$ is the noiseless channel output when the state changes from σ' to σ as a result of input bit v_j.

Equalization with a Linear MMSE Detector

We can represent the Forney model (3.7) in a vectorized form as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} , \tag{3.9}$$

for $\boldsymbol{y} = \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_{N-1} \end{bmatrix}$ and $\boldsymbol{x} = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-1} \end{bmatrix}$. The $N \times N$ matrix $\boldsymbol{H}$ is given as

$$\boldsymbol{H} = \begin{bmatrix} h_0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ h_0 & h_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ h_0 & h_1 & h_2 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & & \vdots \\ 0 & h_0 & h_1 & h_2 & \cdots & h_{L-1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & h_0 & h_1 & h_2 & \cdots & h_{L-1} & 0 \\ 0 & 0 & \cdots & 0 & h_0 & h_1 & h_2 & \cdots & h_{L-1} \end{bmatrix} ,$$

with the assumption that $x_i = 0$ for $i < 0$. The linear MMSE estimator of $\boldsymbol{x}$ computes $\hat{\boldsymbol{x}}$ which minimizes $E\left\{|\hat{\boldsymbol{x}} - \boldsymbol{x}|^2\right\}$.

To do soft equalization however we compute a linear MMSE estimate $\hat{x}_i$ which minimizes $E\left\{|\hat{x}_i - x_i|^2\right\}$. The estimate is given by

$$\begin{aligned} \hat{x}_i &= \mathrm{E}\{x_i\} + \mathrm{Cov}\{x_i, \boldsymbol{y}\}\,\mathrm{Cov}\{\boldsymbol{y}, \boldsymbol{y}\}^{-1}(\boldsymbol{y} - \mathrm{E}\{\boldsymbol{y}\}) \\ &= \mu_i + \sigma_i \boldsymbol{h}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}) , \end{aligned} \tag{3.10}$$

where $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_N + \boldsymbol{H}\boldsymbol{V}\boldsymbol{H}$, $\sigma_i = \mathrm{Cov}\{x_i, x_i\}$ $\mu_i = \mathrm{E}\{x_i\}$, $\boldsymbol{\mu} = \mathrm{E}\{\boldsymbol{x}\}$, $\boldsymbol{h}$ is the i^{th} column of $\boldsymbol{H}$ and $\boldsymbol{V} = \mathrm{diag}\{\sigma_0 \quad \sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_{N-1}\}$. The computational complexity of inverting $\boldsymbol{\Sigma}$ is prohibitive for large N. To get circumvent this, a window of size $W = W_1 + W_2 + 1$ is used instead of the whole sequence. The window covers W_2 symbols before y_i and W_1 symbols after y_i. If W_1 and W_2

are both larger than the memory of the channel L satisfactory performance can be obtained for relatively small values of W with almost no further gains by increasing W. The equations are then modified to be

$$\hat{x}_i = \mu_i + \sigma_i \boldsymbol{h}_i^{\mathsf{T}} \boldsymbol{\Sigma}_i^{-1} \left(\boldsymbol{y}_i - \boldsymbol{H}_i \boldsymbol{\mu}_i \right) . \tag{3.11}$$

Here $\boldsymbol{H}_i$ is $W \times (W+L)$ sub-matrix $\boldsymbol{H}[i-W_2 : i+W_1, i-W_2-L : i+W_1]$ of $\boldsymbol{H}$. With this model we can extract the soft information by using the assumption the estimation error $e_i = \hat{x}_i - x_i$ is independent of x_i and is Gaussian distributed with zero mean. Better approximations of the distribution are possible [11] but results in more complex computation. With this assumption we have that $\hat{X}_i | X = x_i$ is distributed as $\mathcal{N}\left(x_i, \mathrm{Var}(E_i)\right)$, that is,

$$\mathrm{L}(x_i) = \ln \frac{p\left(\hat{x}_i - +1\right)^2 / (2\mathrm{Var}(E_i))}{p\left(\hat{x}_i - -1\right)^2 / (2\mathrm{Var}(E_i))} .$$

3.5.3 Capacity for Channels with ISI

The ultimate limit for a noisy ISI channel with input X and output Y is given by the maximum mutual information between X and Y, maximized over the distribution of X. As mentioned in Section 2.2.3 it is common to work with constrained capacity also commonly called the symmetric information rate (SIR), where the distribution of X is uniform. That is, we wish to compute the information rate

$$I(X;Y) = \lim_{N \to \infty} \frac{1}{N} I(X_1, X_2, \ldots X_N; Y_1, Y_2, \ldots Y_N) , \tag{3.12}$$

where X_i is uniformly distributed among the q possible symbols of the modulation alphabet. We know that

$$I(X;Y) = H(Y) - H(Y|X) .$$

For an ISI channel with additive Gaussian noise $H(Y|X)$ can be computed in closed form but $H(Y)$ can only be computed numerically [41]. The simulation steps are as follows

1. Sample a very long sequences x^N and y^N. That is randomly generate x^N and produce y^N by convolution with the channel impulse response.

2. Compute $p(y^N)$. This is done using the BCJR algorithm as follows; Consider the forward recursion of the BCJR algorithm with

$$\alpha_t(\sigma) = \sum_{\sigma'} \alpha_{t-1}(\sigma') \cdot \gamma_t(\sigma', \sigma) \quad \text{for } t = 1, \ldots N ,$$

$$p(y^N) \text{ is then given as } \quad p(y^N) = \sum_{\sigma'} \alpha_t(\sigma') \text{ for } t = N .$$

   The algorithm is however adjusted by introducing normalization constants to avoid numerical issues.

3. Estimate $\hat{I}(X;Y) = -\frac{1}{N} \log_2 \left(p\left(y^N\right) \right) - H(Y|X) .$

3.6 Classical Code Design for Iterative Receivers

For good performance the code and the detector have to be matched. If one simply chooses a strong code, that is, one which performs very well in an AWGN channel, it does not necessarily imply that the turbo equalizer will work well even if the equalizer is optimal.

A common classical code design approach is to optimize the degree distribution to maximize performance for given channel conditions. This means the transmitter has to know the channel as well as the SNR. Common tools used are EXIT charts and density evolution. We will briefly discuss each of these two.

Code Design for ISI Channels using DE

We can represent a turbo equalizer using factor graphs. Fig. 3.12 shows the a factor graph for turbo equalization with unstructured irregular LDPC code. The detector is denoted by the $\mathcal{H}$ and has noisy observations $\boldsymbol{z}$. The density

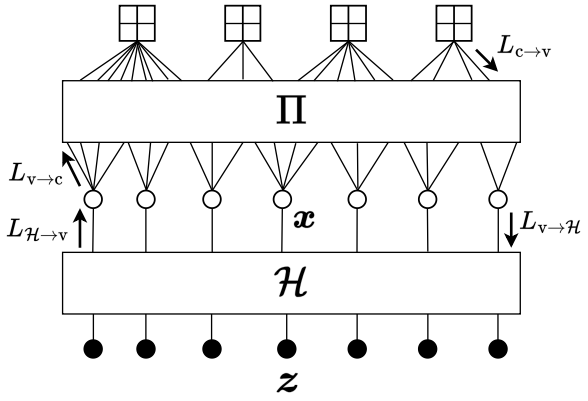


Figure 3.12: Factor graph for an irregular LDPC code and a detector for an ISI channel.

passed from the detector to the decoder, $p(L^{(\ell)}_{\mathcal{H}\to\mathrm{v}})$, is a function of the incoming densities from all VNs $p(L^{(\ell-1)}_{\mathrm{v}\to\mathcal{H}})$ and the noise distribution. Denoting this function as $\mathcal{T}$ we have

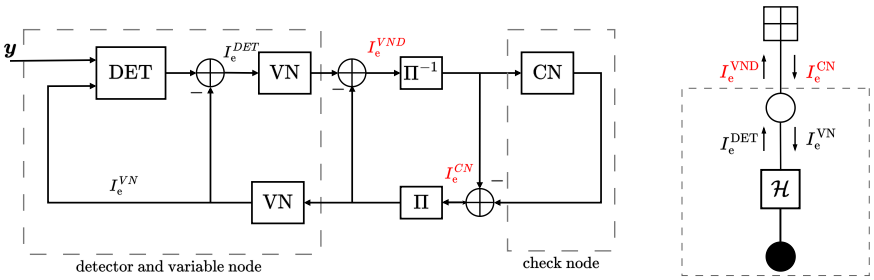
$$p(L^{(\ell)}_{\mathcal{H}\to\mathrm{v}}) = \mathcal{T}\left(p(L^{(\ell-1)}_{\mathrm{v}\to\mathcal{H}}), \sigma\right) .$$

It is not possible to obtain $\mathcal{T}(.,.)$ in closed form for Gaussian noise (it can be computed for erasure noise) but it can be evaluated via Monte Carlo methods. For each VN, its outgoing density to an edge is the convolution of the density $p(L^{(\ell)}_{\mathcal{H}\to\mathrm{v}})$ from the detector and the $d_{\mathsf{v}} - 1$ incoming densities from other edges, where d_{v} is the VN degree. The average density from the VNs to a neighboring CN, $p(L^{(i)}_{\mathrm{v}\to\mathrm{c}})$, is then obtained by averaging over the degree distribution $\lambda(x)$. At each CN of degree d_{c}, the outgoing density is computed from the $d_{\mathsf{c}} - 1$ incoming densities in a nested fashion using a two-dimensional lookup table for discretized density evolution [7]. Similar to the VNs, the average density to a

VN, $p(L_{\mathrm{c}\to\mathrm{v}}^{(i)})$, is obtained by averaging over the degree distribution $\rho(x)$. After I_{C} iterations within the code, the density passed from a VN to the detector, $p(L_{\mathrm{v}\to\mathcal{H}}^{(\ell)})$, is the convolution of the incoming d_{v} densities from the neighboring CNs. The density evolution update equation for the joint BP decoding of the code and channel is thus given as

$$
\begin{aligned}
p(L_{\mathrm{v}\to\mathrm{c}}^{(i)}) &= p(L_{\mathcal{H}\to\mathrm{v}}^{(\ell-1)}) \circledast \boldsymbol{\lambda}\left(p(L_{\mathrm{c}\to\mathrm{v}}^{(i-1)})\right) , \\
p(L_{\mathrm{c}\to\mathrm{v}}^{(\ell)}) &= \boldsymbol{\rho}\left(p(L_{\mathrm{v}\to\mathrm{c}}^{(i-1)})\right) , \\
p(L_{\mathrm{v}\to\mathcal{H}}^{(\ell)}) &= \mathsf{L}\left(L_{\mathrm{c}\to\mathrm{v}}^{(I_{\mathrm{C}})})\right) , \\
p(L_{\mathcal{H}\to\mathrm{v}}^{(\ell)}) &= \mathcal{T}\left(p(L_{\mathrm{v}\to\mathcal{H}}^{(\ell)})\right) .
\end{aligned}
$$

For a given density a, $\boldsymbol{\lambda}(a) = \sum_j \lambda_j a^{\circledast(j-1)}$, $\boldsymbol{\rho}(a) = \sum_j \rho_j a^{\boxed{\circledast}(j-1)}$ and $\mathsf{L}(a) = \sum_i \mathsf{L}_i a^{\circledast(j)}$. The operator $\circledast$ represents the convolution of densities while $\boxed{\circledast}$ represents the density transformations at the CN as used in [34].



(a) (b)

Figure 3.13: (a) Block diagram for the design of LDPC codes for ISI channels. The VN and detector (DET) are combined to a single node (b) The compact graph notation.

Code Design for ISI Channels using EXIT Charts

As described in Section 3.3 EXIT charts are effective tools in classical code design. With turbo equalization with LDPC codes we have three components passing messages, namely the check nodes, the variable nodes and the channel. To obtain a two-dimensional EXIT chart, it is common practice to combine the detector and the variable node into a single entity that interacts with the check node [12]. Fig. 3.13 shows such a figure and the corresponding compact graph notation. The EXIT chart is plotted with a curve for extrinsic information from the combined detector and VN $I_{\mathrm{E}}^{\mathrm{VND}}$ and the extrinsic information from the check node $I_{\mathrm{E}}^{\mathrm{CN}}$. It can be seen form the block diagram that $I_{\mathrm{E}}^{\mathrm{VND}}$ is a function of the channel condition (SNR) and the a priori information to the detector as well the VN degree distribution.

3.6.1 Waterfall versus Error Floor

The performance of a code in iterative systems in terms of BER can be characterized by two regions namely the waterfall region and error floor region. The waterfall region is the relatively low SNR region where the BER starts to fall sharply. This can be predicted from the BP thresholds using EXIT charts (or more accurately using DE). The error floor region, on the other hand, is the region at high SNR where the BER tends to remain relatively constant. Typically, if one chooses a strong code with good MAP threshold the resulting waterfall performance is poor but the error floor is good. On the other hand, if the code is weak the waterfall performance is good but the error floor is bad. Fig. 3.14 illustrates this trade-off using an EXIT chart. Note that the detector's curve

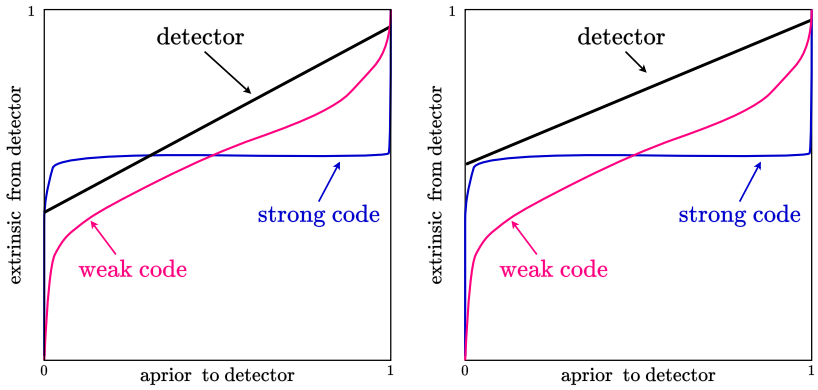


Figure 3.14: Illustration of the trade-off between error floor and water floor using an EXIT chart. In (a) the SNR is low so the strong code does not have a waterfall but the weak code does. In (b) The SNR is increased such to the point where the strong code has waterfall.

moves upwards with increasing SNR but the curves of the codes are not affected. A strong code has a very steep curve at the edges (making it almost flat at the middle) while a weak code has a less steep curve at the edges. Since the waterfall happens when the code and detector do not cross, it can be seen that the strong code will have its waterfall at a higher SNR.

3.6.2 Changing Channel Conditions

Another challenge when designing codes for iterative systems using classical approaches is the changing channel conditions. Since we need to take into account the particular channel conditions in order to optimize the degree distribution when the channel is changed we cannot guarantee good performance. To get the best performance we would need to re-optimize the code to match the new channel conditions. This approach is not practical for scenarios where we expect the channel to change considerably.

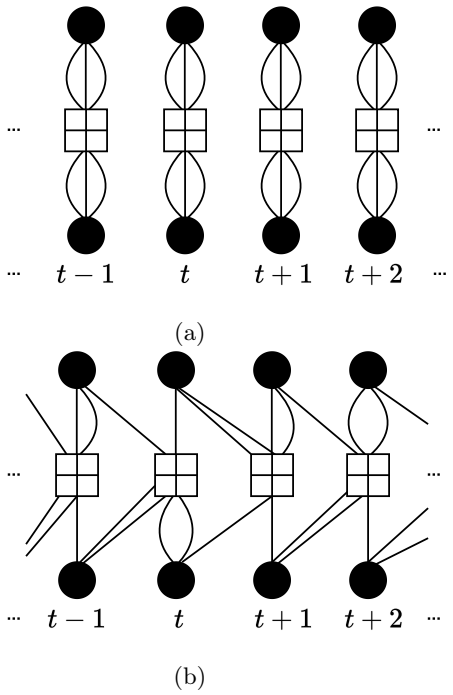


Figure 3.15: (a) Copies of the underlying factor graph (b) SC-LDPC with $w = 1$.

3.7 Spatial Coupling

With spatial coupling, memory is introduced in the factor graph of the iterative reciever such that blocks at different time instants are interconnected. This was first introduced for LDPC codes [13] and later for other classes of codes such as turbo-like codes [42]. Spatially coupled codes have very good performance approaching the capacity for AWGN channels [43]. Furthermore, the performance in the waterfall region does not result in compromised error floor, thus proving useful in avoiding the trade-off between waterfall and error floor performance [16].

We proceed to briefly discuss how spatial coupling is applied in LDPC codes and turbo-likes codes.

3.7.1 Spatially Coupled LDPC Codes

Consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ LDPC code and its corresponding factor graph. The factor graph of an SC-LDPC codes is constructed by placing L copies of the underlying factor graph in L spatial positions in the range $\mathcal{L} \in \{1, \ldots, L\}$. Fig. 3.15(a) shows the compact graph representation of three copies at positions, $t-1$, t, and $t+1$ for a $(3, 6)$ regular code. Each spatial position consists of N VNs, represented by dark circles, and M CNs ($M = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} N$), represented by squares with a cross. The L copies are coupled as follows: each VN at position $t \in \mathcal{L}$ is connected to CNs in the range $[t, \ldots, t+m]$, where m is referred to as the coupling memory. Hence, each CN at position t is connected to VNs in the range $[t-m, \ldots, t]$.

The CNs at the edges have low degree. This makes them more effective at decoding the VNs connected to them. With high probability the VNs at the edges are decoded correctly and can thus be considered known when the decoder decodes VNs deeper in the chain. This results in effective low CN degree for the inner checks. We thus have a wave-like effect spreading in the coupled chain from the edges.

3.7.2 Spatially Coupled Turbo-Like Codes

The concept of spatial coupling is not limited to LDPC codes where the constraint nodes of the factor graph are single parity-check codes. It can also be applied to other scenarios where the constraint node is any other function. Here we introduce spatially coupled turbo-like codes whereby the constraint nodes are convolutional codes which were first explored by Moloudi *et al.* [42, 44]. Examples of such codes are the PCC (simply turbo codes), braided codes and SCC. We focus on the SCC codes as they are closely related to the work in this thesis.

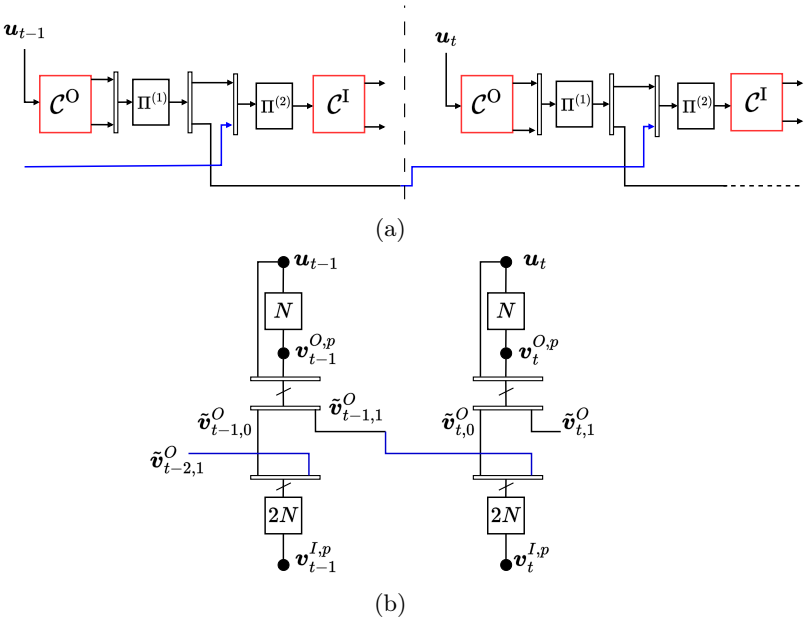


Figure 3.16: (a) Encoder for SC-SCC code with $m = 1$(b) The corresponding compact graph representation.

Spatially Coupled SCC

An SC-SCC with coupling memory m is constructed as follows. We consider systematic encoders for both the outer and inner encoder. Consider a block of information bits $\boldsymbol{u}_t$ which is fed to the outer encoder at time t. We also denote $\boldsymbol{v}_t^{O,p}$ and $\boldsymbol{v}_t^{I,p}$ as the parity bits of the outer and inner encoders, respectively. Furthermore, we denote the code bits at the output of the outer encoder at time t as $\boldsymbol{v}_t^{O} = (\boldsymbol{u}_t, \boldsymbol{v}_t^{O,p})$. The code bits $\boldsymbol{v}_t^{O}$ are permuted by a permutation $\Pi^{(1)}$ to

a sequence $\tilde{\boldsymbol{v}}_t^O$ which is then divided into $m+1$ sequences $\tilde{\boldsymbol{v}}_{t,j}^O (j = 0, \ldots, m)$ of equal length. The input to the inner encoder at time t is the sequence $(\tilde{\boldsymbol{v}}_{t,0}^O, \tilde{\boldsymbol{v}}_{t-1,1}^O, \ldots, \tilde{\boldsymbol{v}}_{t-m,m}^O)$ after it has been reordered by a permutation $\Pi^{(2)}$. The inner encoder then produces the parity bits $\boldsymbol{v}_t^{I,p}$. The code sequence at time t is then $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{O,p}, \boldsymbol{v}_t^{I,p})$. For $t < 0$ we assume that $\boldsymbol{v}_t = 0$. Fig. 3.16(a) shows an encoder for time slots $t-1$ and t for coupling memory $m = 1$. The corresponding compact graph notation is shown in Fig. 3.16(b). For terminated SC-SCC, t takes vales in the range $(0, \ldots, L-1)$, where L is the chain length. The information sequence at the end of the chain is chosen such that $\boldsymbol{v}_t = \mathbf{0}$ for $t > L-1$.

The known bits at the edges ($t < 0$ and $t > L-1$), just like in the LDPC case, make the decoder more effective in decoding the bits at the edges thus creating a wave effect.

3.7.3 Threshold Saturation: a New Paradigm in Code Design

The good performance of spatially coupled codes is explained by a phenomenon known as *threshold saturation*. With threshold saturation the BP threshold of the spatially codes approaches the MAP threshold of the underlying code ensemble. This was first shown in [45] and proved mathematically in [15]. This opens up a new paradigm in code design whereby the MAP threshold now matters in code choice. In the classical approach, iterative receivers are designed by choosing codes whose BP thresholds are better without considering the MAP threshold. This is due to infeasible complexity of implementing a MAP decoder. But many codes with bad BP thresholds have good MAP thresholds and vice versa. Now with threshold saturation we can achieve the good MAP threshold while using a BP decoder with manageable complexity. This however comes with some challenges. One of these challenges is in order to minimize the latency. To achieve this a window decoder is used resulting into some performance loss in terms of waterfall and error floor. But if the window size is big enough we can still have an advantage over classical systems.

More important is the universality of spatial coupling which makes it robust against changes in channel conditions. We can use a single code to achieve performance close to the capacity regardless of the conditions of the channel [46] which is a big advantage compared to the classical systems.

Chapter 4

Group Testing with Sparse Graphs

4.1 Introduction to Group Testing

The aim of group testing (GT) is to identify items which are defective in a population by testing items in groups instead of testing each item individually. For a population of n items with k of them defective the aim is identify these defective items using m tests whereby $m < n$. This can lead to significant reduction in the number of tests compared to testing each item separately especially if the number of defective items is very small compared to n. It was first introduced by Dorfman [19] during World War II for testing army inductees for syphilis. Several blood samples were mixed and tested together. Since syphilis was a rare disease, many tests were negative which meant all the individuals in those tests were ruled as negative resulting into major savings on the number of tests.

The participation of items into tests can be modeled by an $m \times n$ matrix $\boldsymbol{A} = (a_{i,j})$ called the test matrix. The rows of $\boldsymbol{A}$ correspond to the tests while the columns correspond to the items and $a_{i,j} = 1$ if item j participates in test i otherwise $a_{i,j} = 0$. The test assignment can also be represented by a bipartite graph with n variable nodes (VNs) corresponding to items and m check nodes (CNs) corresponding to tests. An edge connects VN j, v_j, and CN i, c_i, if item j participates in test i. In other words, $\boldsymbol{A}$ is the adjacency matrix of the bipartite graph. The matrix in (4.1) below shows the test assignment with $n = 6$ and $m = 3$.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{4.1}$$

The corresponding bipartite graph is shown in Fig. 4.1. In this assignment item 1 participates in tests 1 and 3 while item 2 participates in tests 1 and 2.

For non-quantitative group testing the test results can be either positive (represented by binary value of 1) or negative (represented by binary value of 0). The test result s_i of a given test i is given by the OR operation on all the items

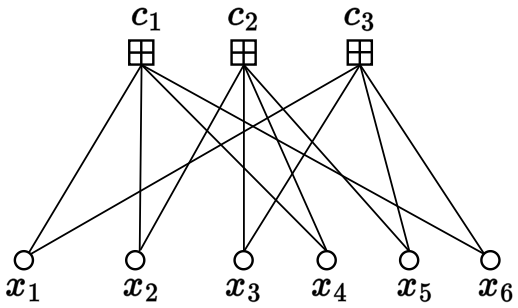


Figure 4.1: Bipartite graph corresponding to the assignment matrix in (4.1).

connected to it. That is

$$s_i = \bigvee_{j=1}^{n} a_{i,j} x_j \,. \tag{4.2}$$

We define the *rate* as the ratio of the number of tests to the number of items[1], that is;

$$\Omega = \frac{m}{n} \,.$$

This notation is different to the definition of rate used in [47] where the rate is defined as the amount of information in bits about the items status X gained per test.

Example 4.1.1. *Consider a GT scheme with the test assignment matrix* $\boldsymbol{A}$ *for 4 tests and 6 items given below with items 1 and 6 being defective.*

$$\begin{matrix} \textcircled{x_1} & x_2 & x_3 & x_4 & x_5 & \textcircled{x_6} & \quad \boldsymbol{s} \end{matrix} \tag{4.3}$$

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} . \tag{4.4}$$

The syndrome $\boldsymbol{s}$ *is shown to the right whereby tests 1 and 3 are negative. This implies that we know for sure that the items connected to them are non-defective i.e* $x_2 = x_3 = x_4 = x_5 = 0$.

On the other hand test 2 is positive which means at least one of x_1, x_2 *and* x_4 *is defective. Since we know that* x_2 *and* x_4 *are not defective,* x_1 *must be defective i.e* $x_1 = 1$.

With test 4 however we have two possibilities now, either $x_6 = 1$ *or* $x_6 = 0$. *This can not be resolved with certainty. What value* x_6 *is assigned depends on the requirement of an application or assumptions made by the decoding algorithm.*

[1] Note that, interpreting $\boldsymbol{A}$ as the parity-check matrix of a code, $\Omega = 1 - R$, where R is the code rate.

4.1.1 Applications of Group Testing

GT has since been applied in many fields apart from testing for rare diseases in medicine. For example it is used in molecular biology for identifying rare genetic conditions [48–50]. In data forensics it is used to identify parts of data which have been tampered with [22, 23]. One-way hash functions are used to generate a hash for the data. This hash is stored and is compared with a new hash run on the data. If the two do not match it implies the data has been tempered with. To know which parts of the data is tempered with we need a hash for each part which can be demanding for more storage. With GT a hash is generated for several parts thus saving memory while maintaining the capacity to identify the tempered parts. Group testing can also be used to detect denial of service attacks. This was demonstrated in [51] and [52] where the server is divided into a number of virtual servers corresponding to tests thus detecting which virtual server receives the greatest traffic and thus determine which users are providing the largest amount of traffic. In wireless communication GT is used in identifying users in multi-access communication [25, 53–55]. In this setting the channels or slots correspond to tests and the devices attempting to access the channel are the items.

4.1.2 Classification of Group Testing

Adaptive vs Non-adaptive

GT can be either adaptive or non-adaptive depending on how the tests are designed. With adaptive GT, the tests are done in stages and the test design at each stage depends on the outcome of the previous stage. For example in the Dorfman case, if the test result in the first stage is positive, all individuals in the pool are tested separately, while a negative result is conclusive and has no followup. For non-adaptive GT however, the tests are all designed in advance. Adaptive GT results in fewer tests than non-adaptive GT but with non-adaptive GT the tests can be run in parallel which makes it attractive for time sensitive applications.

Quantitative vs Non-quantitative

Furthermore we can classify GT as either quantitative or non-quantitative. In quantitative GT each test result shows how many items in the tests are defective while with non-quantitative the results are binary, i.e., one if at least one item is defective and zero otherwise. Some variations exists between these two extremes. For example, we may have a semi-quantitative GT where the test results give a number representing a range of possible number of defective items [56].

Exact vs Partial recovery

If we focus on the success of recovery, GT can be classified as either exact or partial recovery. With an exact recovery criterion, the requirement is that every defective item is correctly classified as defective, and every non defective item is correctly classified as non defective. With partial recovery, however, we may tolerate a small number of incorrectly classified items, perhaps with different allowances for false positives (non defective items incorrectly classified

as defective) and false negatives (defective items incorrectly classified as non defective).

Noiseless vs Noisy testing

In some scenarios the test results can be corrupted to give false positives or false negatives by the testing procedure itself. This is called noisy testing while the case where the test results are not corrupted by the procedure is termed as noiseless testing.

Combinatorial vs. Probabilistic Prior

Classifying according to the distribution of defective items we have combinatorial vs probabilistic prior. With combinatorial prior, there is a fixed number of defectives with the set of defectives being uniformly distributed among the sets of the given size. With probabilistic prior, each item is independently defective with some probability γ. Various generalizations exist where the items status are correlated reflecting some situations like in diseases where people who are in close contact has a higher chance of having the same status.

4.1.3 Non-quantitative Group Testing with Sparse graphs

In this thesis we consider GT where the adjacency matrix is sparse. That is, the number of edges is very small compared to the number of maximum possible $m \times n$. Works with sparse graphs were first introduced in what is called SAFFRON for non-quantitative GT [26] and its improvement using generalized LDPC codes [27]. It was shown that with a simple peeling decoder most of the defective items could be identified with high probability.

4.2 Quantitative GT Based on GLDPC

The work [28] introduced a quantitative group testing scheme based on regular GLDPC codes where the test matrix $\boldsymbol{A}$ corresponds to the parity-check matrix of a GLDPC code. Particularly, the construction in [28] is as follows. Consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph with n VNs and m_{B} CNs and its corresponding $m_{\mathsf{B}} \times n$ adjacency matrix $\boldsymbol{B}$. To construct the test matrix $\boldsymbol{A}$, each of the d_{c} non-zero elements in a row of $\boldsymbol{B}$ is replaced by a column of an $n_{\mathsf{u}} \times d_{\mathsf{c}}$ signature matrix $\boldsymbol{U} = \left(\mathbf{1}_{1\times d_{\mathsf{c}}}^{\top}, \boldsymbol{H}_{\mathsf{t}}^{\mathsf{T}}\right)^{\mathsf{T}}$, where $\mathbf{1}_{1\times d_{\mathsf{c}}}$ is a $1 \times d_{\mathsf{c}}$ all-ones vector and $\boldsymbol{H}_{\mathsf{t}}$, of dimensions $t\log_2(d_{\mathsf{c}}+1) \times d_{\mathsf{c}}$, is the parity-check matrix of a t-error correcting BCH code of length d_{c}. Hence, $n_{\mathsf{u}} = t\log_2(d_{\mathsf{c}}+1)+1$, and the total number of tests is given by $m = m_{\mathsf{B}} n_{\mathsf{u}}$. (Note that, for a GLDPC code-based GT scheme, contrary to the bipartite graph in Fig. 4.1, each of the CNs corresponds to a bundle of n_{u} tests.)

The rate for the construction in [28] is

$$\Omega = \frac{m}{n} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}\Big(t\left\lceil\log_2(d_{\mathsf{c}}+1)\right\rceil + 1\Big), \tag{4.5}$$

where the ceiling function $\lceil . \rceil$ takes care of cases where $d_{\mathsf{c}}+1$ is not a power of two.

Example 4.2.1. *Consider the case with a regular* $(2, 4)$ *adjacency matrix* $\boldsymbol{B}$ *with* $n = 14$ *VNs and* $m_{\mathsf{B}} = 4$ *CNs defined as*

$$\boldsymbol{B} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix},$$

and a 1-error correcting BCH code whose parity-check matrix is defined as

$$\boldsymbol{H}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The signature matrix is then given as

$$\boldsymbol{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

This makes the test matrix to be

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Decoding to identify the defective items is performed via peeling decoding, where, at each iteration, due to the t-error correcting capability of the BCH codes, a CN connected to t or less unresolved defective items can identify them and their adjacent edges are peeled off the graph. This is repeated until all defective items are identified or there is no test with t or less defective items.

The decoding of each CN is done by first identifying if a test that has t or less defective items by looking at the first row of $\boldsymbol{U}$. For those tests with t or less defective items, the decoder proceed to determine the location of defective items by using the Berlekamp–Massey algorithm [29].

The performance of quantitative GT with the GLDPC scheme does not improve by increasing the strength of the BCH code. It actually degrades after $t = 2$ for most of the range of γ. Fig. 4.2 shows a plot of the minimum number of tests required to successfully resolve all defective items. It can be seen

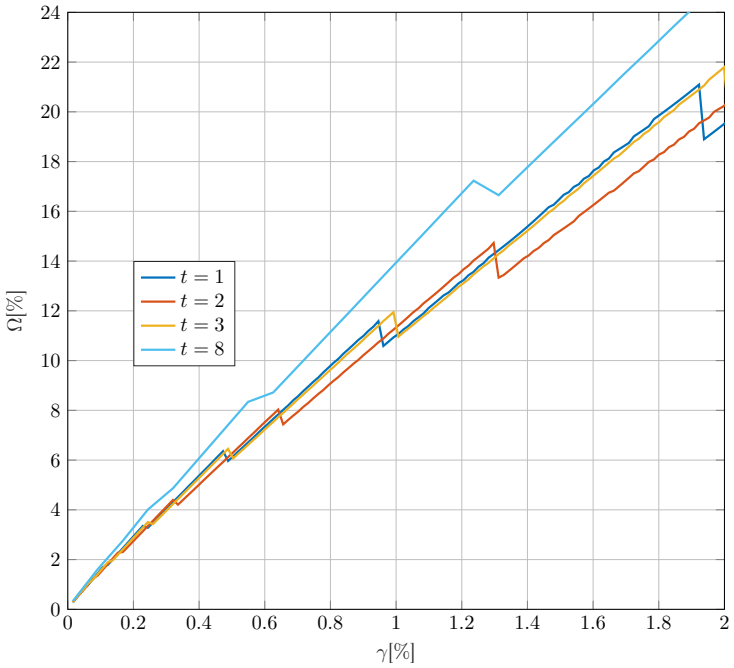


Figure 4.2: Minimum rate Ω required to identify all defective items for different values of γ using the GLDPC scheme.

that for higher values, like $t = 8$, the performance much worse than that for small values of ts. This seems contrary to expectations at first glance, as one would expect that at the same rate a stronger error correction would provide better performance. In this thesis we investigate why this is the case providing an explanation for this and an alternative construction which has better performance.

Chapter 5

Summary and Contributions

In this chapter the main results of the included papers are summarized. Furthermore general conclusions and outlook of further research questions are discussed. This is done for each of the two parts, ISI channels and group testing.

5.1 Part I: Channels with Intersymbol Interference

5.1.1 Research Contributions

Paper I: Spatial Coupling in Turbo Equalization

In this paper we investigate the application of spatial coupling in turbo equalization for channels with ISI using fine length simulations and EXIT chart analysis. We show that choosing an off-the-shelf strong code results in poor waterfall performance but good error floor performance, while using a weak code results in the opposite namely good waterfall but bad error floor. This behaviour is due to the sub-optimally of the BP decoding algorithm and not the codes themselves. That is, if we were to use an optimal decoder the strong code will also have good waterfall in addition to the good error floor.

We apply spatial coupling and demonstrate that we can have both good waterfall and good error floor while using the BP algorithm. We apply coupling using three different schemes. In the first scheme, we couple between the encoder and the channel, resulting into waterfall performance beyond the threshold predicted by EXIT charts. In the second scheme we use a spatially coupled code which results in significant improvement compared to coupling between the encoder and the channel. In the third we use a spatially coupled code and couple between the encoder and the channel. This results in further improvement which is not so significant when compared to using only a spatially coupled code.

Furthermore we investigate the effect of using spatially coupled codes with sub-optimal linear MMSE equalizer. It is shown that with spatial coupling the performance of a linear MMSE equalizer is close to the optimal MAP equalizer thus becoming attractive to use especially when the channel memory is large.

Paper II: On the Universality of Spatially Coupled LDPC Codes over Intersymbol Interference Channels

In this paper we investigate the universality of spatially coupling in channels with ISI. We do this using density evolution for both channels with erasures and AWGN.

For channels with erasures, we derive the exact input/output transfer function for a general ISI channel. This adds a richer range of channels to the study of ISI channels with erasures beyond the dicode channel which was the only one available in literature. We then compute the symmetric information rate for three different channels and the BP and MAP thresholds using regular LDPC codes with increasing node degree.

We show with examples how spatial coupling results universally achieves the SIRs of three different channel using a single LDPC code.

Paper III: Robust Performance Over Changing Intersymbol Interference Channels by Spatial Coupling

In this paper we investigate the practical implication of the universality of spatial coupling for channels with ISI. We do this using density evolution thresholds and finite length simulations using both the MAP equalizer and the LMMSE equalizer. In particular we compare spatial coupling with classical code design using irregular LDPC codes.

With classical code design the degree distribution is optimized for a particular channel using EXIT charts or density evolution. This results in good performance for the particular channel designed for but the performance degrades severely when the channel is changed. With spatial coupling however, one regular code with high node degree performs close to the SIR for different ISI channels. We thus have not only good performance but more importantly robustness against changing channel conditions.

We also show that with spatial coupling the same code performs well even if the channel detector is changed. That is, we have good performance even if the detector is changed form MAP to LMMSE which is not the case with classical code design.

5.1.2 General Conclusions

In this thesis we investigated the application of spatial coupling to channels with intersymbol interference. We particularly investigated how it performs well in comparison to classical code designs especially when the channel conditions or detector type is changed. This was for the AWGN case and erasures. To achieve this, we derived the input/output transfer function for the a general BEC thus expanding the available channels for studying ISI channels with erasures which were only studied for the dicode channel.

We show that with spatial coupling we could avoid the trade-offs between the performance in the waterfall region vs the performance in the error floor region which is often encountered when choosing codes for iterative receivers for ISI channels. With spatial coupling we could choose a strong code and obtain good performance for both the waterfall and error floor region.

We also managed to demonstrate that with spatial coupling we can achieve the symmetric information rate (i.u.d capacity) of an ISI channel. More importantly this performance is robust against changing channel conditions. This robustness gives spatial coupling an advantage over classical methods like optimizing codes by tweaking the degree distribution to get the desired performance. These optimization works but are limited to the particular channel, that is we can approach close to the capacity of a given channel we optimizing for but if the channel is changed to another one the performance degrades substantially.

5.1.3 Further Research

Some of the questions which remain to investigated for iterative receivers for channels with ISI include the following:

- The effect of accuracy of channel state information on the equalization with and without spatial coupling. We conjecture that with spatial coupling the requirements on the accuracy of the channel estimation can be relaxed but more work has to be done on this to verify this.
- Another area for further study would be faster-than-Nyquist (FTN) signaling whereby the ISI is created intentionally by packing more pulses than the Nyquist limit in order to increase bandwidth efficiency. With FTN there is usually a limit on the compression factor beyond which reliable communication even with iterative receivers cannot work well. We conjecture that with spatial coupling this limit can be pushed further thus getting higher bandwidth efficiency with low modulation order. Furthermore since the channels impulse response can be long some shortening techniques are usually employed. Here spatial coupling can be investigated providing a room for better performance with simplified schemes such as channel shortening.

5.2 Part II: Group Testing with Sparse Graphs

5.2.1 Research Contributions

Paper IV: Low-Density Parity- Check Codes and Spatial Coupling for Quantitative Group Testing

In this paper we investigated non-adaptive group testing (GT) based on sparse graphs. It has been shown in previous works that using sparse graphs substantially improves the performance of group testing. For example the use of generalized LDPC (GLDPC) codes with t-error correcting BCH codes has shown a significant gain in quantitative GT. The scheme relies on the local error correction capability of BCH codes to resolve defective items. In the paper we propose a simpler algorithm with simple LDPC codes which have no local error correcting capability. The scheme utilizes two extreme cases; one being when all items in a tests are non-defective and the other when all items in a test are defective. We derive exact density evolution equations for the proposed LDPC scheme and compute the corresponding thresholds. The scheme performs much better than the GLDPC scheme. This is a result of that with the GLDPC codes we need a bundle of tests to correct a single error (to correct t errors we need

tests proportional to t). This cost in number of tests is not counteracted by the local error correction capability.

We compare the schemes performance using DE thresholds and finite length simulations. We look at two perspectives of performance; in one the proportion of defective items is fixed and we determine the minimum rate (the ratio of number of tests to population size) required for successful detection. This is the standard perspective in GT literature. We introduce another perspective whereby the rate is fixed and determine the largest proportion of defective items which can be successfully recovered. This perspective is more in line with the DE in channel codes where the graph structure is assumed fixed and some channel parameter is varied.

Furthermore, we apply spatial coupling to both the GLDPC and LDPC based scheme. Both schemes improve with coupling but the GLDPC does not gain much compared to the LDPC scheme. The DE tended to converge to a stable value suggesting threshold saturation. We do not, however ,prove this formally in this paper.

Paper V: Spatially Coupled LDPC and GLDPC Codes for Quantitative Group Testing

In paper IV we presented a quantitative GT using LDPC codes and applied spatial coupling. The DE thresholds suggested a convergence to a limit suggesting threshold saturation.

In this paper we prove that threshold saturation indeed happens in quantitative GT with LDPC codes presented in Paper IV. We show this by proving that the density evolution recursion for quantitative GT using LDPC codes satisfies the conditions of being a vector admissible system. We note that among the two perspective of performance evaluation introduced in Paper IV, the perspective where the proportion of defectives is fixed is a vector admissible system while the other perspective does not meet the requirements. We could however demonstrate that the two perspective are related making it possible to map the potential thresholds from one perspective to the other.

We also proved threshold saturation for the existing GT with GLDPC codes. The proof in this case was done by showing that the scheme satisfies the criteria for being a scalar admissible system. For both schemes we computed the potential threshold and verified that the coupled thresholds approach the potential thresholds.

We also investigated why the GLDPC based scheme does not perform well compared to the LDPC scheme even with spatial coupling. We showed that this is a result of the limitations of the bounded minimum decoder used by the scheme. We proposed modifications on the decoder with one modification utilizing the LDPC decoder we proposed in Paper IV.

Paper VI: LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet

This paper focuses on the Quantitative GT with LDPC codes as introduced in paper IV aiming at improving the decoder. It can be observed that the decoder in paper IV is a hard decision decoder thus one could postulate that a soft massages decoder would improve the performance. The complexity of such

a decoder can however be prohibitive. We propose another approach where the items are grouped into bundles of q items. These bundles can then take values from 0 to q thus we have artificially introduced non-binary alphabets. Motivated by works on counter braids, we design a decoder which bounds the bundles providing lower and upper bounds. These bounds shrink with iterations thus becoming more close with the actual value of the bundle. With a few additional tests accepting items as the conventional (i.e not in bundles) LDPC based scheme we can then successfully resolve all the items.

We derive the density evolution equations for the proposed decoder. The performance of the decoder is then compared with the conventional LDPC scheme through DE thresholds and finite length simulations. The results show that the proposed decoder shows considerable improvement over the conventional LDPC scheme in paper IV.

5.2.2 General Conclusions

In this thesis we investigated non-adaptive quantitative GT using sparse graphs. We proposed a scheme using simple LDPC codes which does not have local error correction. This scheme was analysed by deriving their DE equations and computing the thresholds which were compared to schemes widely proposed in the literature which uses generalized LDPC code with BCH component codes. The BCH codes provide a local error correction capability which results in good performance. The provided LDPC scheme does not have local error correcting capability but results in a significantly better performance than the GLDPC scheme.

We also investigated the reasons for the performance of the GLDPC scheme concluding that it is caused mainly by the decoding algorithm. That is, bounded minimum distance decoder is very far from the optimal decoder. We provide an alternative decoder utilizing the LDPC decoder with improved results.

We further investigated the application of spatial coupling in Quantitative GT. Both the GLDPC and LDPC based schemes benefits from coupling with the LDPC scheme showing more gain thus further consolidating the benefits of the scheme. As in many other scenarios, this benefit from spatial coupling comes from the phenomenon of threshold saturation which was formally proved for GT in this thesis.

It was further shown that the proposed LDPC scheme can be improved by introducing hidden nodes in the graph representing the test assignment. These hidden nodes are introduced by grouping items into bundles of a chosen size. This construction, together with a simple bounding decoder inspired by works in counter braids, results in improved performance without significant complexity increase.

5.2.3 Further Research

Some interesting research questions which can be explored in group testing include

- Quantitative group testing with noisy tests. Questions include what noise models to use and what codes perform best for a given noise model? What about threshold saturation proof for noisy quantitative group testing?

- Quantitative GT using sparse graphs with short block lengths. It remains unclear how cycles affect the performance of GT. What construction criteria should be used to optimize performance with short block length ? How does the performance scale with population size.
- The performance of the GLDPC based GT can be further investigated looking for better ways to construct the component codes. As shown in the thesis, the BCH code and its corresponding minimum distance decoder can be further improved by decoding them differently. We conjecture that better performance can be obtained by constructing the component code with other parameters of consideration than the bounded minimum decoder.

References

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[2] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int Conf. on Communications (ICC)*, Geneva, Switzerland, May 1993.

[4] K. Gracie and M.-H. Hamon, "Turbo and turbo-like codes: Principles and applications in telecommunications," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1228–1254, 2007.

[5] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[6] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics letters*, vol. 33, no. 6, pp. 457–458, 1997.

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

[8] R. McEliece, D. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 140–152, 1998.

[9] D. A. Shnidman, "A generalized Nyquist criterion and an optimum linear receiver for a pulse modulation system," *The Bell System Technical Journal*, vol. 46, no. 9, pp. 2163–277, 1967.

[10] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[11] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 920–952, 2011.

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

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

[14] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

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

[16] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes: A new trade-off between waterfall and error floor," *IEEE Trans. Comm*, vol. 67, no. 5, pp. 3114–3123, 2019.

[17] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.

[18] A. Yedla, H. D. Pfister, and K. R. Narayanan, "Universality for the noisy slepian-wolf problem via spatial coupling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, July-Aug. 2011, pp. 2567–2571.

[19] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[20] A. J. Macula, "Probabilistic nonadaptive group testing in the presence of errors and dna library screening," *Annals of Combinatorics*, vol. 3, pp. 61–69, 1999.

[21] N. J. A. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. W. S. Chan, "Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs," in *IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 2007.

[22] M. T. Goodrich, M. J. Atallah, and R. Tamassia, "Indexing information for data forensics," in *Applied Cryptography and Network Security.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 206–221.

[23] A. J. Macula and L. J. Popyack, "A group testing method for finding patterns in data," *Discrete Applied Mathematics*, vol. 144, no. 1-2, pp. 149–157, 2004.

[24] J. Hayes, "An adaptive technique for local distribution," *IEEE Trans. Comm.*, vol. 26, no. 8, pp. 1178–1186, 1978.

[25] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 185–191, 1985.

[26] K. Lee, R. Pedarsani, and K. Ramchandran, "SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July 2016.

[27] A. Vem, N. T. Janakiraman, and K. R. Narayanan, "Group testing using left-and-right-regular sparse-graph codes," *CoRR*, vol. abs/1701.07477, 2017. [Online]. Available: http://arxiv.org/abs/1701.07477

[28] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *2019 IEEE Inf. Theory Work. (ITW)*, 2019.

[29] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications 2nd Ed.*, ser. Pearson education. Pearson-Prentice Hall, 2004.

[30] E. D. Bloch, *Proofs and Fundamentals: A First Course in Abstract Mathematics 2nd Ed."*. Springer, 2001.

[31] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

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

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

[34] T. Richardson and R. Urbanke, *Modern Coding Theory*. USA: Cambridge University Press, 2008.

[35] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[36] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Comm*, vol. 49, no. 10, pp. 1727–1737, 2001.

[37] C. Measson, A. Montanari, T. J. Richardson, and R. Urbanke, "The generalized area theorem and some of its consequences," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4793–4821, 2009.

[38] G. Ungerboeck, "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems," *IEEE Trans. Inf. Comm.*, vol. 22, no. 5, pp. 624–636, 1974.

[39] G. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 363–378, 1972.

[40] F. Rusek, M. Loncar, and A. Prlja, "A comparison of Ungerboeck and Forney models for reduced-complexity ISI equalization," in *Proc. IEEE Global Telecom. Conference (GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 1431–1436.

[41] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Info. Theory*, vol. 52, no. 8, pp. 3498–3508, 2006.

[42] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, 2017.

[43] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[44] Moloudi, Saeedeh, "Spatially Coupled Turbo-Like Codes," Ph.D. dissertation, Lund University, 2018.

[45] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, 2010.

[46] M. Lentmaier, I. Andriyanova, N. Hassan, and G. Fettweis, "Spatial coupling - a way to improve the performance and robustness of iterative decoding," in *European Conference on Networks and Communications (EuCNC)*, Paris,France, June-July 2015.

[47] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: An information theory perspective," *Foundations and Trends in Com. and Info. Theory*, vol. 15, no. 3-4, pp. 196–392, 2019.

[48] R. N. Curnow and A. P. Morris, "Pooling dna in the identification of parents," *Heredity*, vol. 80, no. 1, pp. 101–109, 1998.

[49] J. Tammisola, R. Akerman, S. Lapinjoki, and V. Kauppinen, "Strategies of pooling for parentage analyses applying dna markers," *Biometrics in Plant Breeding: Applications of Molecular Markers*, pp. 186–194, 1994.

[50] Y. Erlich, A. Gilbert, H. Ngo, A. Rudra, N. Thierry-Mieg, M. Wootters, D. Zielinski, and O. Zuk, "Biological screens from linear codes: theory and tools," *BioRxiv*, p. 035352, 2015.

[51] S. Khattab, S. Gobriel, R. Melhem, and D. Mosse, "Live baiting for service-level dos attackers," in *IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, April 2008.

[52] Y. Xuan, I. Shin, M. T. Thai, and T. Znati, "Detecting application denial-of-service attacks: A group-testing-based approach," *IEEE Trans. Parallel and distributed systems*, vol. 21, no. 8, pp. 1203–1216, 2009.

[53] A. Graell i Amat, G. Liva, E. Paolini, and C. Stefanovic, "Coded slotted aloha with stopping set resolution: A group testing approach," in *Asilomar Conference on Signals, Systems, and Computers*, 2022, pp. 677–681.

[54] T. Berger, N. Mehravari, D. Towsley, and J. Wolf, "Random multiple-access communication and group testing," *IEEE Trans. Comm.*, vol. 32, no. 7, pp. 769–779, 1984.

[55] B. S. Tsybakov, “Resolution of a conflict of known multiplicity,” *Problems of Information Transmission*, vol. 16, no. 2, pp. 69–82, 1980.

[56] A. Emad and O. Milenkovic, “Semiquantitative group testing,” *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4614–4636, 2014.

[57] P. Damaschke, "Threshold group testing" in General Theory of Information Transfer and Combinatorics (Lecture Notes in Computer Science), Berlin, Germany:Springer-Verlag, vol. 4123, pp. 707-718, 2006.

Part II

Included Papers

Paper I

Spatial Coupling In Turbo Equalization

In this paper we consider spatial coupling in turbo equalization and demonstrate that the code design trade-off between the performance in waterfall and error floor regions can be avoided. We introduce three coupling schemes and compare their performances, where the first method introduces coupling between the encoder and the channel, while the second uses a spatially coupled (SC) code. In the third scheme we use both a coupled code and couple between the code and the channel. We show by computer simulations that, with spatial coupling, we can have good performance in both the error floor and the waterfall region with reasonable decoding latency by using a window decoder. We show this for both the maximum *a posteriori* (MAP) and linear minimum mean square (MMSE) equalizers.

Keywords: Spatial Coupling, Inter-Symbol interference, Equalization, LMMSE Equalizer.

1 Introduction

Turbo equalization has been shown to be effective in mitigating the effect of intersymbol interference (ISI) by having the equalizer and decoder exchange soft information iteratively rather than each component working separately [1], [2]. This iterative exchange is an instance of belief propagation (BP) and can be analyzed by factor graphs [3]. An optimal receiver, however, is a MAP detector of the transmitted symbols taking into account the joint effect of the code constraints and the ISI channel.

The choice of codes for a turbo equalization usually involves a trade-off between the performance in the waterfall versus the performance in the error floor region. Choosing a weak code results in good waterfall performance but bad error floor, while choosing a strong code results in a bad waterfall performance but good error floor. This trade-off however is a result of the BP decoding process and not the codes themselves, that is, if we use a joint MAP detector with a sufficiently large codeword length the strong code would result in a better waterfall performance approaching the MAP threshold of the combined factor graph.

Spatially coupled low-density parity-check (SC-LDPC) codes have been shown to exhibit threshold saturation, whereby the BP threshold of the coupled ensemble approaches the MAP threshold of the uncoupled ensemble, [4], [5], [6]. In [7] it was proved that threshold saturation also occurs in spatially coupled turbo-like codes. Furthermore, [8] outlined a new trade-off between error floor and waterfall performance. In particular, it was shown that when spatially coupled, serially concatenated codes (SCCs) can have both better waterfall and error floor performance than parallel concatenated codes (PCCs). This is despite the fact that SCCs have a poorer waterfall performance than PCCs when not coupled.

SC-LDPC codes were also investigated in coded modulation [9]. It was observed that with spatial coupling the performance of the codes was less sensitive to the chosen type of mapping, thus demonstrating some universality behavior with spatial coupling. In [10], spatial coupling between the code and detector for faster-than-Nyquist signaling and coded modulation was investigated. The output block of an encoder is split such that the input to the detector is a combination of various sub-blocks at different times. The work investigated the best way to split the output from the encoder to optimize convergence with the fewest number of iterations. Binary erasure channels with memory were studied in [11], [12]. It was shown that with SC-LDPC codes, threshold saturation also occurs in this channel. In [12] it was also shown empirically that SC-LDPC codes exhibit threshold saturation in an ISI channel with AWGN.

In this paper, we first demonstrate the challenge involved in the choice of codes by examining simulation results with a simple convolutional code versus a 5G LDPC code as component codes and explain the design trade-off using extrinsic information transfer (EXIT) charts. We then consider the application of spatial coupling in three different ways in turbo equalization to show that this trade-off can be avoided. In the first scheme, we couple between the encoder and channel leaving the code uncoupled. Then we derive an SC-LDPC code from the 5G LDPC code and use it without any coupling at the channel and lastly we couple both at the code and channel level. We show that for both MAP and MMSE equalizers, the spatially coupled code results in a larger gain

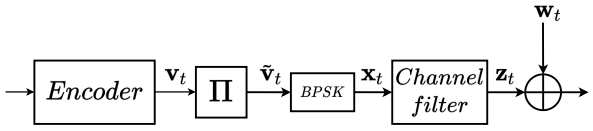


Figure 1: Block diagram showing the transmitter and the ISI channel.

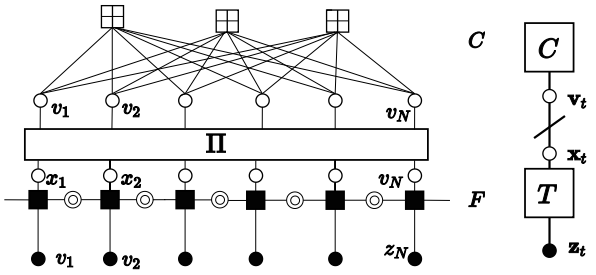


Figure 2: (a) Combined factor graph of an LDPC code and channel without coupling (b) compact factor graph representation.

than coupling at the channel input, while coupling both components is only slightly better than the coupled code alone. We hence managed to show that with spatial coupling we can use a strong code and obtain best performance in both the waterfall and error floor regions.

2 Turbo Equalization

A model for the transmitter is shown in Fig. 1 where a block of K information bits, $\mathbf{u}_t$, is encoded by a code of rate $R_c = K/N = 1/2$ to produce N code bits $\mathbf{v}_t$. These are then permuted to a new sequence $\tilde{\mathbf{v}}_t$ which is mapped to symbols $\mathbf{x}_t$ using binary phase shift keying (BPSK) modulation. The input symbol sequence $\mathbf{x}_t$ passes through the ISI channel filter with discrete impulse response $\mathbf{h}$ to obtain the output $\mathbf{z}_t$. The received symbol sequence is the sum of $\mathbf{z}_t$ and AWGN $\mathbf{w}_t$. The channel has $M_h + 1$ taps $h[0], \ldots, h[M_h]$, where M_h is the channel memory. Throughout this paper we use the following channel from [13]:

$$\mathbf{h} = \begin{bmatrix} 0.277 & 0.46 & 0.688 & 0.46 & 0.277 \end{bmatrix} . \tag{1}$$

The system can be represented by a factor graph, which shows the relationship between variables in the system [14]. As an example, Fig. 2 shows a factor graph of a regular $(3, 6)$ LDPC code and a channel. Black circles represent variables which have noisy observations at the receiver while white circles represent variables which are not observed at the receiver. State variables are represented by double circles and square nodes represent constraints in the code or channel. Following the notation in [8] we can represent the factor graph in a compact form by introducing node types and represent variables of equivalent distributions by a single node. For example, neglecting the edge effects we can represent a block of input symbols to the channel by a single node $\mathbf{x}_t$ as shown in Fig. 2(b) since the messages along connected edges have the same distribution. In the Figure, code constraints are denoted by $\mathcal{C}$ while the channel is denoted by $\mathcal{H}$.

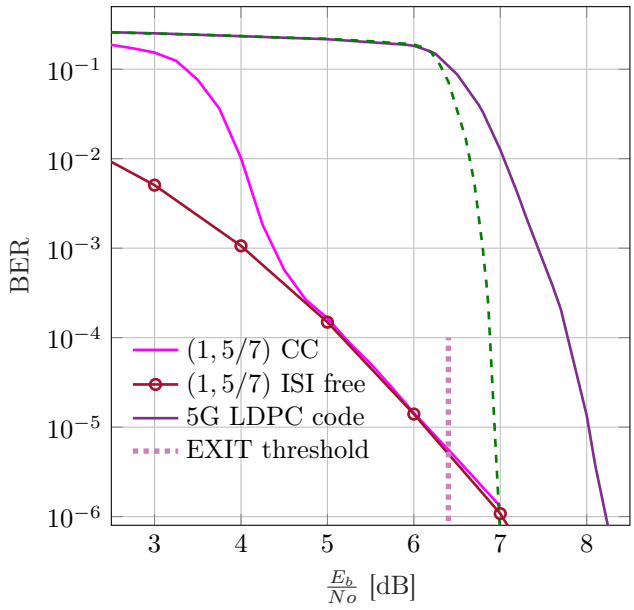


Figure 3: Comparison of turbo equalization using a MAP equalizer with LDPC code and a convolutional code demonstrating the trade-off of choosing a weak versus a strong code. For both codes $N = 5120$. The dashed line shows the 5G code with a permutation of length 51200.

To get good results in terms of bit error rate (BER) with relatively low complexity, the equalizer and decoder exchange information in a number of iterations. This iterative equalization and decoding is often called turbo equalization [2]. The MAP equalizer is an optimal equalizer and is implemented by a trellis following the work in [2]. Its complexity however grows as 2^{qM_h}, making it impractical when the memory of the channel or the modulation order (q) is large. A linear MMSE equalizer, on the other hand, though less accurate does not suffer from this exponential growth in complexity. The linear equalizer is implemented using a window approach [2].

The choice of the code usually involves a trade-off between the performance in the waterfall region and the error floor region. Using a weak code gives good waterfall performance but results in poor error floor while if we choose a code which is strong in an AWGN channel, it results in poor waterfall performance but good error floor. Two codes are used to illustrate this, a $(1, 5/7)$ systematic convolutional encoder (representinga weak code) and a 5G LDPC code (representing a strong code). Both codes have block length $N = 5120$ and the 5G code is obtained from base graph BG2 by lifting the graph by 256 as detailed in [15]. The number of iterations between the code and channel is 8 in all scenarios, while for the LDPC code we use 30 iterations within the code with a parallel schedule. It can be seen in Fig. 3 that when $\mathcal{C}$ is a convolutional code (CC) the waterfall performance is good but the error floor is bad, limited by the performance of the code in an ISI free case. While when $\mathcal{C}$ is the LDPC code it shows very poor waterfall performance. For example the BER is still above 10^{-1} at an SNR of 6 dB while for the convolutional code it is close to 10^{-5}.

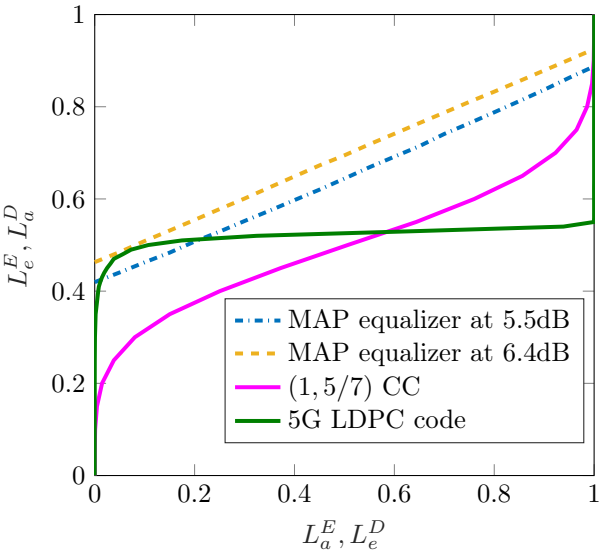


Figure 4: EXIT chart predicting the performance of 5G LDPC code and $(1, 5/7)$ CC in turbo equalization.

This trade-off can be explained by observing the EXIT charts [16] for both codes and the channel in Fig. 4. The EXIT curve of the channel shifts up with increasing SNR, while those of the codes do not vary with SNR as the codes have no direct observations. The LDPC code has a nearly flat inverse EXIT function, which makes it intersect the channel at points of low mutual information (hence higher BER) for all SNR values below 6.4 dB, while the convolutional code being a weaker code has a shape which makes it intersect the channel at points of higher mutual information thus resulting in good waterfall performance. The waterfall performance of the 5G code can be improved slightly by increasing the permutation length as shown by the dashed line in Fig. 3 but it can not exceed the EXIT threshold shown by the vertical dotted line.

As a solution to this problem, a method using irregular convolutional codes, optimized together with precoders for a desired waterfall performance is proposed in [3]. A drawback of such a solution is that it depends much on the channel and the equalizer type thus making it unsuitable in changing channel conditions. It can also result in bad error floors due to the weak component codes used in the optimization. Furthermore, since there is a limit to the choice of precoders determined by the memory of the channel [3], for some channels like the one we chose for this case the use of precoders does not show significant improvement in the performance. It is also not possible to use a precoder with linear equalizers without increasing the decoding complexity, which can be done with MAP equalizers [3].

But this trade-off is not inherent in the system itself but rather in the decoding process. If we were able to use a joint MAP decoder for both the code and the channel, the LDPC code would outperform the convolutional code in both waterfall and error floor regions. Spatially coupled codes have been shown to exhibit threshold saturation, where the BP threshold of the coupled codes

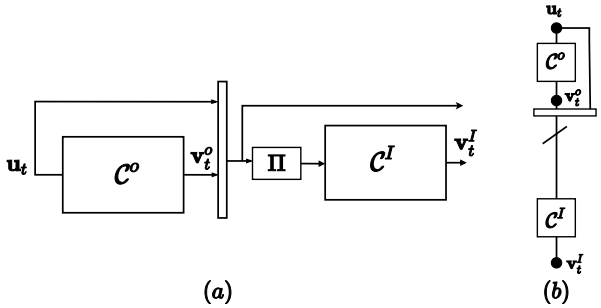


Figure 5: (a) Block diagram of a SCC encoder (b) Compact graph representation.

approaches the MAP threshold of the underlying uncoupled codes.

3 Spatial Coupling in Turbo Equalization

With spatial coupling, memory is introduced in the factor graph of the turbo equalization system such that blocks at different time instants are interconnected. Three options are discussed.

3.1 Coupling between Encoder Output and Channel

In [8] it was shown that the waterfall performance of serially concatenated codes (SCC) is improved significantly when they are spatially coupled. Comparing the factor graph of a SCC in Fig. 5 and that of the channel and code in Fig. 2, we observe that the two systems are equivalent. The channel acts as an inner code, $\mathcal{C}^I$, with rate 1 and non-binary outputs $\mathbf{z}_t$ which have noisy observations, while the symbols from the outer component code (equivalent to $\mathcal{C}^o$) are not transmitted which can be viewed as punctured symbols in the corresponding SCC code. Thus we can also couple the output of the code and the channel in a fashion similar to the one applied in [8] as follows.

Consider a normal system with the code and channel without any coupling. In this setting each block $\mathbf{v}_t$ of coded bits is permuted and put into the channel as shown in Fig. 6(a). In Fig. 6(b) each block of permuted bits $\tilde{\mathbf{v}}_t$ is split into two sub-blocks of equal lengths. One sub-block is put to the channel at time t while the other sub-block is connected another sub-block produced at time $t+1$. In this way we are effectively introducing blockwise memory between the code and the channel. The memory in this case is $m = 1$, since we need output from one previous block in the past in order to find the current input to the channel.

In general, for a coupled system with memory m, a block of N code bits $\tilde{\mathbf{v}}_t$ produced by the encoder at time t after interleaving by the permutation Π^1, is split into $m+1$ sub-blocks $\tilde{\mathbf{v}}_{t,0}, \ldots, \tilde{\mathbf{v}}_{t,m}$ each having $\frac{N}{m+1}$ bits. The input to the channel at time t is a sequence of symbols from the set $\{\tilde{\mathbf{v}}_{t-m,m}, \ldots, \tilde{\mathbf{v}}_{t,0}\}$ after being permuted by a second permutation Π^2. This is repeated for $t = 0 \ldots L-1$, where L is the length of the chain. The code bits $\tilde{\mathbf{v}}_t$ are set to zero for $t < 0$ and $t > L - m - 1$. This introduces known bits at the beginning and at the end

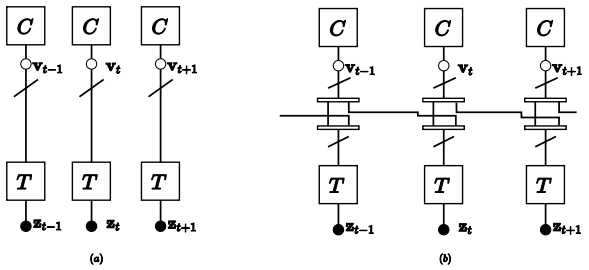


Figure 6: Compact graph representation of (a) channel and code without coupling (b) Coupling between the encoder and channel. (In both figures the symbol $\mathbf{x}_t$ is not shown.)

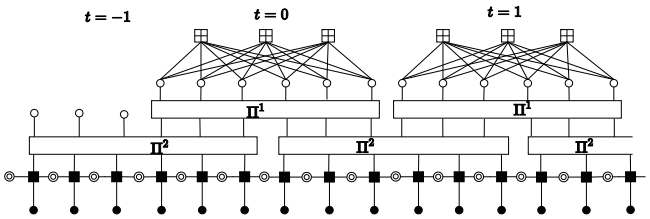


Figure 7: Factor graph representation of coupling between code and channel for $m = 1$.

of the chain incurring a rate loss which becomes negligible as L becomes large, as it can be seen in the rate of the coupled system given by

$$\begin{aligned} R &= \frac{K(L-m)}{NL} \\ &= R_c \left(1 - \frac{m}{L}\right) \ , \end{aligned} \tag{2}$$

which approaches the rate of the code R_c as L grows. These known bits, however, play an important role in improving performance of the belief propagation decoder as discussed in Subsection 3.4. Fig. 7 shows the factor graph of such a system at the start of the chain.

3.2 Using a Spatially Coupled Code

We can also use a spatially coupled code as the component code. We use a spatially coupled LDPC code constructed as described in [17]. Coupling is done on the protograph, followed by lifting the protograph by some chosen lifting factor and some permutations. With coupling memory $m = 1$, a variable node v_i with degree d_{v_i} splits its d_{v_i} edges into check nodes at time t and $t + 1$. This splits the base matrix $\mathbf{B}$ into two sub-matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ such that $\mathbf{B}_0 + \mathbf{B}_1 = \mathbf{B}$. The chain is terminated after L sections. The base matrix of the

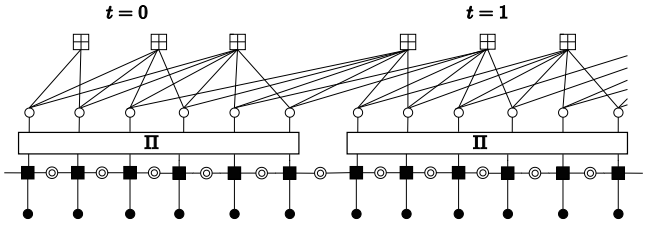


Figure 8: Factor graph representation of turbo equalization with a coupled LDPC code.

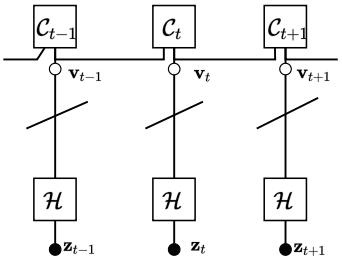


Figure 9: Compact graph representation of turbo equalization with a spatially coupled code.

terminated convolutional protograph is given by

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ & \mathbf{B}_1 & \ddots & \\ & & \ddots & \mathbf{B}_0 \\ & & & \mathbf{B}_1 \end{bmatrix}. \tag{3}$$

For simplicity we illustrate this in Fig. 8 with a regular (3,6) code with base matrices

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

In this scheme every block of M bits at time t is permuted, mapped to symbols and sent over the channel. Fig. 9 shows the corresponding compact graph representation. This scheme is used in order to exploit the advantages of window decoding as elaborated in Section 3.4, especially reduced latency, as it makes it possible to decode a block without waiting for the whole chain to be received.

3.3 Coupling at both the Code and the Channel Level

Looking at the overall factor graph, we can see that it is possible to use a spatially coupled code and at the same time coupling the encoder output blocks,

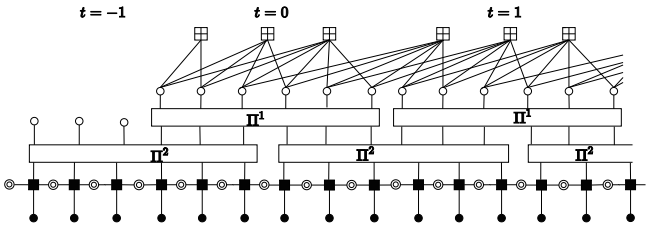


Figure 10: Factor graph representation of turbo equalization with coupling at both the encoder and the channel level.

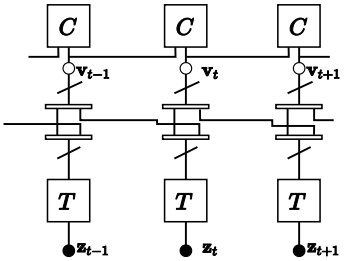


Figure 11: Compact graph representation of coupling both the code and channel.

thus having a graph as shown in Fig. 10, whereas Fig. 11 shows the compact graph. The encoding of the code is not affected by the memory introduced at the input of the channel which is done as in Section 3.1.

3.4 Window Decoding of a Coupled System

In order to get good results with coupling and minimize the latency it is essential to use window decoding. Since blocks which are m or more apart are not affecting each other directly and the effect further decays with increasing distance we can decode a block at time t by considering blocks within a window W, with $W \geq m + 1$ [18]. With the scheme introducing memory between the

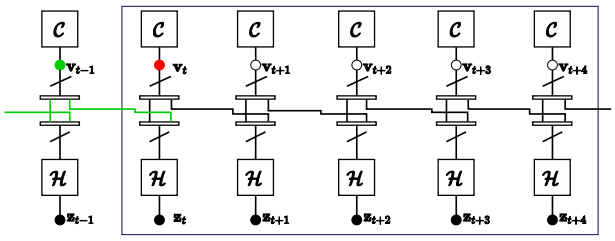


Figure 12: Window decoder with $W = 5$, decoding block t with coupling between the code and channel only. The green dots represents decoded blocks of bits and the red ones the block being decoded.

code and the channel, as depicted in Fig. 12, the fist input block to the channel at time t contains known bits which corresponds to log-likelihood ratios (LLRs) with large magnitudes. These known bits result in improved estimates which

are passed on to the code as good extrinsic information which can now correct more errors. As the exchange is repeated the code can correct more and more errors and spread the good beliefs in the window through the interconnections between blocks. For the SC-LDPC codes, the almost known bits are provided by the low-degree check nodes at the beginning of the chain in $\mathbf{B}_0$, which provide more reliable messages.

As the window moves to the next block, the elements from the first block are now mostly known and the process repeats itself. With window decoding the effect of known bits can spread into the graph with less complexity compared to a scheme which would involve the whole chain, as such a system will require a lot more iterations to spread the effect in the graph. Furthermore, the latency is reduced as we can decode a block without waiting for the whole chain to be received. Once we have waited for the first W blocks to decode the first block, subsequent blocks can be decoded after reception of only one more block. One drawback of the window decoding scheme is error propagation, since errors in one block can, in rare cases, affect all subsequent blocks. Solutions to this drawback are suggested in [19], [20] but in this paper we use window decoding without any modifications.

4 Performance Analysis

The performance of the different forms of coupling is analyzed through computer simulations. For all types of coupling we consider $L = 100$ and the decoder uses $W = 5$. The capacity limit shown is the constrained capacity, where the input to the channel is restricted to be identically and uniformly distributed (i.u.d) and is computed numerically as described in [21]. Using the MAP equalizer we can see in Fig. 13 that coupling at the channel alone, results in a gain of about 2 dB for a BER below 10^{-5}. It is interesting to note that this occurs below the EXIT threshold of the code of 6.4 dB, which can not be exceeded by the code alone even if we use a very long interleaver. When a spatially coupled code is used we observe a larger gain of around 4 dB with 1 dB gap to the i.u.d capacity limit. The gain is 2 dB more than the case with coupling at the channel alone, but it comes at the cost of changing the code and thus the encoder and decoder. Using both a spatially coupled code and coupling at the channel results in very small gain (about 0.05 dB) when compared to using a spatially coupled code alone.

When a linear MMSE equalizer is used (see Fig. 14), we observe a similar trend as in the MAP case. Coupling at the channel shows a gain of about 5 dB while the coupled code shows a gain of around 8.5 dB. The gains in each case are higher than their MAP counterparts. As a result of this the coupled code with linear equalizer is only 1 dB away from that of the MAP case, as opposed to 5 dB difference when no coupling is applied.

These increased performance comes at the cost of increased complexity at the decoder as each block (except at the boundaries) is visited 5 times (the window size) making the effective number of iterations five times more than the uncoupled case.

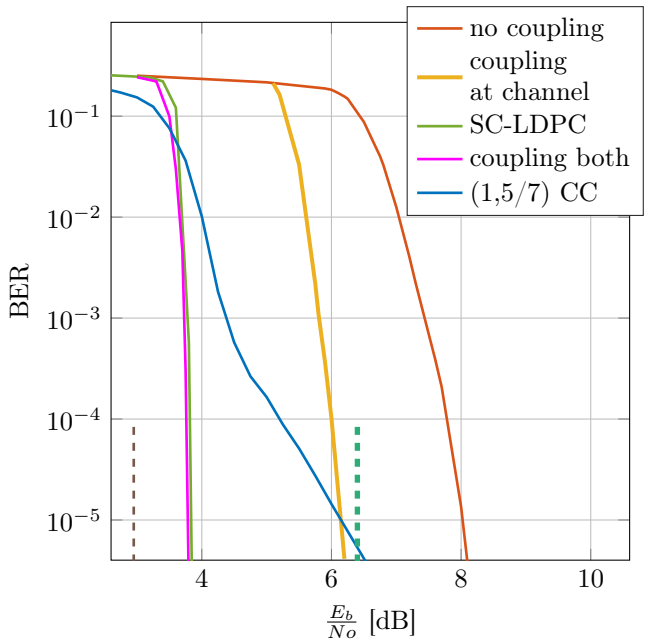


Figure 13: The effect of different types of coupling using a MAP equalizer. The vertical dashed lines at 2.9 dB and 6.4 dB mark the i.u.d capacity limit and the uncoupled EXIT threshold, respectively.

5 Conclusions

We illustrated three ways of coupling in turbo equalization and analyzed their performance. We showed that with spatial coupling the trade-off between the performance in the waterfall versus the error floor region in the choice of codes can be avoided. We also showed that with spatial coupling we can get good performance with both MAP or linear MMSE equalizers. Furthermore spatial coupling is superior to other approaches in the design of iterative receivers, which require a code adaptation to the particular channel, making them impractical in changing channel conditions. By using window decoding, the improved performance can be obtained with relatively low latency.

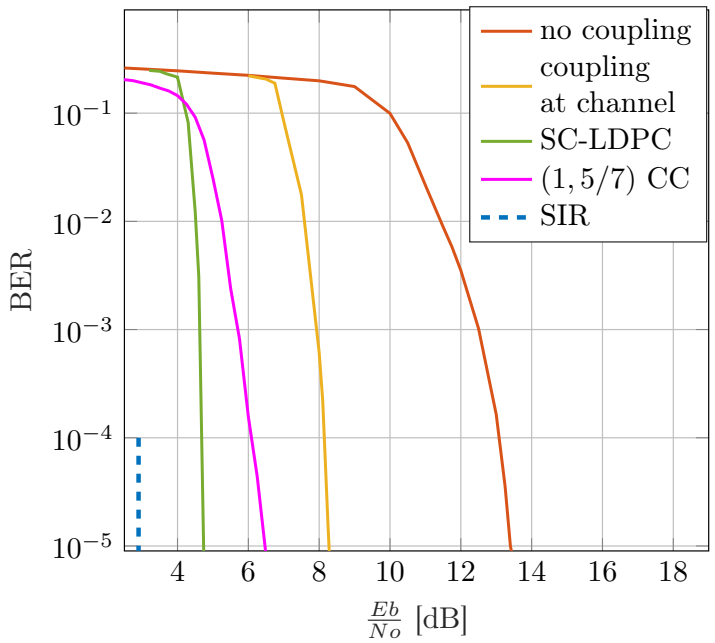


Figure 14: Simulation results using MMSE equalizer showing the effect of coupling at the channel and using a coupled LDPC code.

References

[1] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[2] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[3] M. Tüchler, "Turbo equalization," Ph.D. dissertation, Technical University Munich, 2005.

[4] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, 2010.

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

[6] A. Yedla, Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, 2014.

[7] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct 2017.

[8] ——, "Spatially coupled turbo-like codes: A new trade-off between waterfall and error floor," *IEEE Trans. Comm.*, vol. 67, no. 5, pp. 3114–3123, May 2019.

[9] L. Schmalen and S. ten Brink, "Combining spatially coupled LDPC codes with modulation and detection," in *International ITG Conference on Systems, Communication and Coding (SCC)*, Munich, Germany, Jan. 2013, pp. 1–6.

[10] T. Benaddi, C. Poulliat, and R. Tajan, "A general framework and optimization for spatially-coupled serially concatenated systems," in *Proc. IEEE Global Telecom. Conference (GLOBECOM)*, Singapore, Dec. 2017

[11] S. Kudekar and K. Kasai, "Threshold saturation on channels with memory via spatial coupling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, July-Aug. 2011, pp. 2562–2566.

[12] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *Proc. IEEE Int Conf. on Communications (ICC)*, June 2012, pp. 2181–2186.

[13] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 1987.

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

[15] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee, "An overview of channel coding for 5G NR cellular communications," *APSIPA Transactions on Signal and Information Processing*, vol. 8, p. e17, 2019.

[16] J. Hagenauer, "The EXIT chart - introduction to extrinsic information transfer in iterative processing," in *European Signal Processing Conference*, Vienna, Austria, Sept. 2004, pp. 1541–1548.

[17] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[18] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Braided convolutional codes with sliding window decoding," *IEEE Trans. Comm.*, vol. 65, no. 9, pp. 3645–3658, 2017.

[19] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Combating error propagation in window decoding of braided convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA,June 2018, pp. 1380–1384.

[20] M. Zhu, D. Mitchell, M. Lentmaier, and D. Costello, Jr., "Decoder error propagation mitigation for spatially coupled LDPC codes," in *International Symposium on Information Theory and Its Applications, ISITA*, 2020.

[21] D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, 2006.

Paper II

On the Universality of Spatially Coupled LDPC Codes over Intersymbol Interference Channels

In this paper, we derive the exact input/output transfer functions of the optimal a-posteriori probability channel detector for a general ISI channel with erasures. Considering three channel impulse responses of different memory as an example, we compute the BP and MAP thresholds for regular spatially coupled LDPC codes with joint iterative detection and decoding. When we compare the results with the thresholds of ISI channels with Gaussian noise we observe an apparent inconsistency, i.e., a channel which performs better with erasures performs worse with AWGN. We show that this anomaly can be resolved by looking at the thresholds from an entropy perspective. We finally show that with spatial coupling we can achieve the symmetric information rate of different ISI channels using the same code.

Keywords: Spatial Coupling, Inter-Symbol interference, Equalization, Threshold Saturation, Universality.

---

1 Introduction

Spatial coupling is a powerful concept that improves the belief propagation (BP) decoding threshold of the coupled system to the maximum a-posteriori (MAP) decoding threshold of the underlying uncoupled system. Spatial coupling was initially introduced in the context of low-density parity-check (LDPC) codes [1, 2] and subsequently applied to other classes of codes [3, 4] and scenarios beyond the realm of coding [5].

Threshold saturation was first proven for spatially-coupled LDPC (SC-LDPC) codes for transmission over the binary erasure channel [6] and later for the general class of binary memoryless symmetric channels [7]. Threshold saturation of SC-LDPC codes for transmission over channels with memory was addressed in [8] for intersymbol-interference (ISI) channels with general noise model. Particularly, the authors derived the BP generalized extrinsic information transfer (GEXIT) curves of the corresponding uncoupled systems, from which the MAP threshold can be estimated. The computation of the exact GEXIT curves requires knowledge of the input/output transfer functions of the a-posteriori probability channel detector, which are in general not available in closed form. Thus, one needs to resort to Monte Carlo methods to provide an estimate. For the particular case of the dicode channel with erasures —known also as the dicode erasure channel (DEC)—, in [9] Pfister derived the corresponding transfer function. This was used in [8] to obtain the exact density evolution equations for SC-LDPC codes over the DEC, which are needed for the computation of the GEXIT curves. The authors then showed numerically that threshold saturation occurs for the DEC.

In this paper, we consider SC-LDPC codes for general ISI channels. Our main contribution is the derivation of the transfer functions for general ISI channels with erasures, which allows us to derive the exact density evolution equations for these channels. We then use these equations to compute BP and MAP thresholds for SC-LDPC codes over these channels. The numerical results show that, for large enough coupling memory, threshold saturation occurs for all considered channels. Furthermore, by increasing the Tanner graph density, we show that the BP thresholds approach the symmetric information rates (SIRs) of the corresponding channels, supporting the conjecture in [8] that SC-LDPC codes can universally approach the SIR of ISI channels. We further consider SC-LDPC codes over ISI channels with additive white Gaussian noise (AWGN), which reveal the same behavior.

2 System Model

The system model under consideration is shown in Fig. 1. A binary information sequence $\boldsymbol{u}$ is first encoded by an SC-LDPC code onto codeword $\boldsymbol{v}$. We consider binary transmission, with mapping $0 \mapsto +1$ and $1 \mapsto -1$, over an ISI channel with either erasures or AWGN. In both cases, the modulated sequence $\boldsymbol{x}$ is first transmitted over an ISI channel. The sequence at the output of the ISI filter is denoted by $\boldsymbol{z}$, whose elements take values on a finite alphabet $\mathcal{Z}$. For the ISI-erasure channel, each element of $\boldsymbol{z}$ is erased with probability ε. In this case, the elements of the received sequence, $\boldsymbol{y}$, take values on the finite alphabet $\{\mathcal{Z} \cup ?\}$, with symbol ? denoting an erasure. For the ISI-AWGN channel, $\boldsymbol{z}$ is corrupted

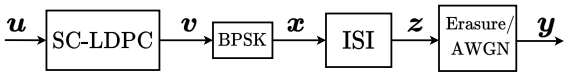


Figure 1: Block diagram showing the transmitter and the ISI channel.

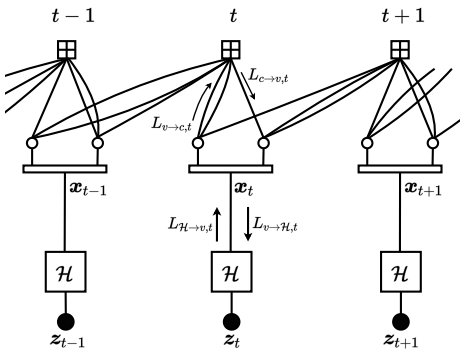


Figure 2: Compact graph representation for equalization with a (3,6) SC-LDPC code with coupling memory $m = 1$.

by AWGN, and the elements of $\boldsymbol{y}$ take on real values.

At the receiver, we consider joint iterative decoding and channel detection, usually referred to as turbo equalization. In particular, we consider BP decoding of the SC-LDPC code and optimal channel detection, performed using the BCJR algorithm over the corresponding trellis. Graphically, decoding is performed over the factor graph depicted in Fig. 2, which combines the Tanner graph of the SC-LDPC code (upper part) and the trellis of the ISI channel (lower part). Specifically, the factor graph is constructed by placing L copies of a $(d_{\mathsf{v}}, d_{\mathsf{c}})$ regular LDPC code of variable node (VN) degree d_{v} and check node (CN) degree d_{c} in L spatial positions in the range $\mathcal{L} \in \{1, \ldots, L\}$. Fig. 2 shows the factor graph for three spatial positions, $t-1$, t, and $t+1$. Each spatial position consists of N VNs, represented by empty circles, and M CNs, represented by squares with a cross. The L copies are coupled as follows: each VN at position $t \in \mathcal{L}$ is connected to CNs in the range $[t, \ldots, t+m]$, where m is referred to as the coupling memory. Hence, each CN at position t is connected to VNs in the range $[t-m, \ldots, t]$. The trellises of the ISI channel at each spatial position are represented by a square labeled with the letter $\mathcal{H}$, referred to as factor node. The VNs represented by the black circles at the bottom of the figure correspond to the symbol sequences at the output of the ISI channel (prior to the addition of the erasures or AWGN), denoted by $\{\boldsymbol{z}_t\}$. Note that $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L)$ (see Fig. 1). The rectangles at each spatial position between the Tanner graph of the SC-LDPC code and the channel factor nodes represent multiplexers that multiplex the N code bits (VNs) at each spatial position into a single binary sequence ($\boldsymbol{x}_t$, with $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$) at the input of the channel.

Decoding is then performed by iteratively passing messages along the edges of the graph in Fig. 2.

3 Input/Output Transfer Function of the BCJR Detector for an ISI Channel with Erasures

In this section, we derive the transfer functions of the BCJR channel detector for arbitrary ISI channels with erasures. These transfer functions characterize the output extrinsic erasure probabilities—from the channel detector to the SC-LDPC decoder—as a function of the a-priori erasure probabilities and the channel erasure probability. Particularly, we follow a similar approach to that in [4, 10] for binary convolutional codes over the binary erasure channel, which scales well with the channel memory. Compared to the case of binary convolutional codes, however, the nonbinary alphabet of general ISI channels and the lack of symmetry, which precludes assuming that the all-zero codeword is transmitted, makes the derivation for a bit more complex.

We consider an ISI channel with memory ν and trellis states $s_1, s_2, \ldots, s_{2^\nu}$. Let $\boldsymbol{y}$ and $\boldsymbol{v}$ be the received vector (affected by erasures) and the output of the ISI filter, respectively. Note that for ISI channels with erasures the messages exchanged between the decoder and the channel detector take values on the ternary alphabet $\{+1, -1, ?\}$. We denote by $\hat{\boldsymbol{x}}$ the message vector from the decoder to the channel detector.

We define the forward and backward state metric vectors at time τ, $\tau = 1, \ldots, n$, where n is the length of the ISI channel trellis, as $\boldsymbol{\alpha}_\tau = (\alpha_\tau(s_1), \ldots, \alpha_\tau(s_{2^\nu}))$ and $\boldsymbol{\beta}_\tau = (\beta_\tau(s_1), \ldots, \beta_\tau(s_{2^\nu}))$, respectively. Note that $\boldsymbol{\alpha}_\tau$ and $\boldsymbol{\beta}_\tau$ are probability vectors. In the case of erasures, the vectors $\boldsymbol{\alpha}_\tau$ and $\boldsymbol{\beta}_\tau$ take values on a finite set. The sets of values that vectors $\boldsymbol{\alpha}_\tau$ and $\boldsymbol{\beta}_\tau$ can take on are denoted by $\mathcal{M}_\alpha = \{\boldsymbol{m}_\alpha^{(1)}, \ldots, \boldsymbol{m}_\alpha^{(|\mathcal{M}_\alpha|)}\}$ and $\mathcal{M}_\beta = \{\boldsymbol{m}_\beta^{(1)}, \ldots, \boldsymbol{m}_\beta^{(|\mathcal{M}_\beta|)}\}$, respectively, of cardinality $|\mathcal{M}_\alpha|$ and $|\mathcal{M}_\beta|$. For the particular case of the DEC, which has memory one and the impulse response given in Table 1 (CH-I), there are two possible states, $s_1 = +1$ and $s_2 = -1$, and $\mathcal{M}_\alpha = \mathcal{M}_\beta = \{(1, 0), (0, 1), (0.5, 0.5)\}$. Note that, in general, $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ may differ.

The sequences $\ldots, \boldsymbol{\alpha}_{\tau-1}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_{\tau+1}, \ldots$ and $\ldots, \boldsymbol{\beta}_{\tau+1}, \boldsymbol{\beta}_\tau, \boldsymbol{\beta}_{\tau-1}, \ldots$ form each a Markov chain, which can be properly described by a probability transition matrix, denoted by $\boldsymbol{M}_\alpha$ and $\boldsymbol{M}_\beta$, respectively, where the element (i, j) of $\boldsymbol{M}_a$, $i \in \{1, \ldots, |\boldsymbol{M}_a|\}$, is the probability of transition from state $\boldsymbol{m}_a^{(i)}$ to state $\boldsymbol{m}_a^{(j)}$, with $a \in \{\alpha, \beta\}$.

Our aim is to derive $\boldsymbol{M}_\alpha$ and $\boldsymbol{M}_\beta$ for an arbitrary ISI channel with erasures. We provide some details for the forward recursion. Let $X \in \{+1, -1\}$, $Z \in \mathcal{Z}$, and $Y \in \{\mathcal{Z} \cup ?\}$ be the random variable corresponding to the symbol at the input of the ISI filter, the output of the ISI filter, and the received symbol for a given trellis section. Further, let $\hat{X} \in \{+1, -1, ?\}$ be the random variable corresponding to the incoming message from the decoder. The corresponding realizations are x, z, y, and $\hat{x}$. We denote by ε the channel erasure probability, i.e., the probability that Y is an erasure, and by δ the average probability of erasure of a message from the decoder to the channel detector, i.e., the probability that $\hat{X}$ is an erasure.

Define $P(\boldsymbol{m}_\alpha^{(j)} | \boldsymbol{m}_\alpha^{(i)})$ as the probability of transition from state $\boldsymbol{m}_\alpha^{(j)}$ to state $\boldsymbol{m}_\alpha^{(i)}$ in the forward recursion. This probability can be written as

Table 1: Discrete impulse responses of the considered ISI channels

CH-I	$\boldsymbol{h} = [\ 0.7071 \quad -0.7071\ ]$	$\nu = 1$
CH-II	$\boldsymbol{h} = [\ 0.408 \quad 0.816 \quad 0.408\ ]$	$\nu = 2$
CH-III	$\boldsymbol{h} = [\ 0.227 \quad 0.46 \quad 0.688 \quad 0.46 \quad 0.227\ ]$	$\nu = 4$

Table 2: Possible observations for DEC channel CH-I with $\alpha_t{=}\boldsymbol{m}_\alpha^1$

$\hat{X}, Y$	0,0	0, ?	?, 0	1, −2	1, ?	?, −2	?, ?
α_{t+1}	$\boldsymbol{m}_\alpha^1$	$\boldsymbol{m}_\alpha^1$	$\boldsymbol{m}_\alpha^1$	$\boldsymbol{m}_\alpha^2$	$\boldsymbol{m}_\alpha^2$	$\boldsymbol{m}_\alpha^2$	$\boldsymbol{m}_\alpha^3$
Prob	$\frac{1}{2}\bar{\delta}\bar{\varepsilon}$	$\frac{1}{2}\bar{\delta}\varepsilon$	$\frac{1}{2}\delta\bar{\varepsilon}$	$\frac{1}{2}\bar{\delta}\bar{\varepsilon}$	$\frac{1}{2}\bar{\delta}\bar{\varepsilon}$	$\frac{1}{2}\bar{\delta}\varepsilon$	$\delta\varepsilon$

$$\begin{aligned} P(\boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(i)}) &= \sum_{\hat{x},y} P(\hat{x}, y, \boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(i)}) \\ &= \sum_{\hat{x},y} P(\hat{x}|\boldsymbol{m}_\alpha^{(i)}) P(y, \boldsymbol{m}_\alpha^{(j)}|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) \,, \end{aligned} \tag{1}$$

where we used the shorthand notation $P(\hat{x}, y) = P(\hat{X} = \hat{x}, Y = y)$.

Now let $\mathcal{Z}_{ij}$ be the set of all possible values that Z can take in the transition between state $\boldsymbol{m}_\alpha^{(i)}$ and state $\boldsymbol{m}_\alpha^{(j)}$. Then we have

$$\begin{aligned} P(y, \boldsymbol{m}_\alpha^{(j)}|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) &= \sum_{z \in \mathcal{Z}_{ij}} P(y, \boldsymbol{m}_\alpha^{(j)}, z|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) \\ &= \sum_{z \in \mathcal{Z}_{ij}} P(z, \boldsymbol{m}_\alpha^{(j)}|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) P(y|z) \,. \end{aligned} \tag{2}$$

This follows from the fact that Y is independent of $\{\hat{X}, \alpha_t, \alpha_{t+1}\}$ given Z. Clearly, $P(Y = ?|z) = \varepsilon$ and $P(Y = z|z) = 1 - \varepsilon$.

To compute $P(z, \boldsymbol{m}_\alpha^{(j)}|x, \boldsymbol{m}_\alpha^{(i)})$, we need to consider two cases. If $\hat{X}$ is an erasure, we have $P(z, \boldsymbol{m}_\alpha^{(j)}|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) = P(z, \boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(i)})$ for all $z \in \mathcal{Z}_{ij}$, since Z is no longer constrained by $\hat{X}$. On the other hand, if $\hat{X}$ is not an erasure, we get $P(z, \boldsymbol{m}_\alpha^{(j)}|\hat{x}, \boldsymbol{m}_\alpha^{(i)}) = P(z, \boldsymbol{m}_\alpha^{(j)}|x, \boldsymbol{m}_\alpha^{(i)})$, where $P(z, \boldsymbol{m}_\alpha^{(j)}|x, \boldsymbol{m}_\alpha^{(i)})$ can be zero for some combinations of $\{z, x, \boldsymbol{m}_\alpha^{(i)}\}$. In summary, we obtain

$$P(\hat{x}, y, \boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(i)}) = \begin{cases} \frac{1}{2}\bar{\delta}\bar{\varepsilon} P(z, \boldsymbol{m}_\alpha^{(j)}|x, \boldsymbol{m}_\alpha^{(i)}) & x, z \\ \frac{1}{2}\bar{\delta}\varepsilon & x, ? \\ \delta\bar{\varepsilon} P(z, \boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(i)}) & ?, z \\ \delta\varepsilon & ?, ? \end{cases}, \tag{3}$$

where we used the shorthand notation $\bar{\delta} \triangleq 1 - \delta$ and $\bar{\varepsilon} \triangleq 1 - \varepsilon$.

In Table 2, we give the probabilities $P(\hat{x}, y, \boldsymbol{m}_\alpha^{(j)}|\boldsymbol{m}_\alpha^{(1)})$ in (3) for the DEC. For this channel, whose impulse response is given in Table 1 (CH-I), $\mathcal{Z} = \{-2, -1, 0, +1, +2\}$ (when no normalization by $1/\sqrt{2}$ is applied to the channel taps). The complete probability transition matrix $\boldsymbol{M}_\alpha$ is given by

Table 3: Transfer functions of ISI channels

CH-I	$g(\delta,\varepsilon)=\frac{4\varepsilon^2}{(\delta\varepsilon-\delta+2)^2}$
CH-II	$g(\delta,\varepsilon)=\frac{4\varepsilon^3(4\delta\varepsilon-4\delta-\delta^2\varepsilon+\delta^2+4)}{(\delta^2\varepsilon^3-\delta^2\varepsilon^2+2\delta\varepsilon^2-2\delta+4)^2}$
CH-III	$g(\delta,\varepsilon)$ see bottom of the page

$$\boldsymbol{M}_\alpha=\begin{bmatrix}\frac{1}{2}(1-\delta\varepsilon) & \frac{1}{2}(1-\delta\varepsilon) & \delta\varepsilon\\ \frac{1}{2}(1-\delta\varepsilon) & \frac{1}{2}(1-\delta\varepsilon) & \delta\varepsilon\\ \frac{1}{2}-\frac{\delta}{4}(1+\varepsilon) & \frac{1}{2}-\frac{\delta}{4}(1+\varepsilon) & \frac{\delta}{2}(1+\varepsilon)\end{bmatrix}.$$

Similarly, we obtain the backward recursion probability transition matrix as

$$\boldsymbol{M}_\beta=\begin{bmatrix}\frac{1}{2}\bar{\varepsilon} & \frac{1}{2}\bar{\varepsilon} & \varepsilon\\ \frac{1}{2}\bar{\varepsilon} & \frac{1}{2}\bar{\varepsilon} & \varepsilon\\ \frac{1}{4}\bar{\varepsilon}\bar{\delta} & \frac{1}{4}\bar{\varepsilon}\bar{\delta} & \varepsilon+\frac{1}{2}\delta-\frac{1}{2}\delta\varepsilon\end{bmatrix}.$$

Now, denote the steady state distribution vector of the forward and backward Markov chain by $\boldsymbol{\pi}_a$, which can be computed as the solution to

$$\boldsymbol{\pi}_a=\boldsymbol{M}_a\cdot\boldsymbol{\pi}_a\,, \tag{4}$$

where $a\in\{\alpha,\beta\}$. Also, define the $|\mathcal{M}_\alpha|\times|\mathcal{M}_\beta|$ matrix $\boldsymbol{T}$ with entries

$$T_{ij}=p(\tilde{X}=?|\alpha_t=\boldsymbol{m}_\alpha^{(i)},\beta_{t+1}=\boldsymbol{m}_\beta^{(j)})\,,$$

where $\tilde{X}$ is the message passed from the channel detector to the decoder. In words, T_{ij} is the average (extrinsic) probability that the symbol passed by the channel detector to the decoder is an erasure given α_t and β_{t+1}. Then, the extrinsic erasure probability of a message from the channel detector to the decoder is given by

$$g(\delta,\varepsilon)=\boldsymbol{\pi}_\alpha\cdot\boldsymbol{T}\cdot\boldsymbol{\pi}_\beta\,,$$

which we refer to as the input/output transfer function of the channel detector.

In Table 3, we give the transfer function of three ISI channels of memory $\nu=1$ (CH-I), 2 (CH-II), and 4 (CH-III) with erasures, where CH-I corresponds to the DEC. The impulse response for these channels is given in Table 1.

4 BP and MAP Thresholds for SC-LDPC Codes over ISI Channels

The receiver applies joint iterative message passing detection and decoding between the channel and the decoder, in terms of log-likelihood ratios (L-values) as illustrated in Fig. 2. At iteration ℓ and spatial position t, first the messages $L^{(\ell)}_{\mathcal{H}\to\mathrm{v},t}$ from the channel detector to the variable nodes are updated with

$$\frac{\varepsilon^5\left(-\delta^6\varepsilon^6(\varepsilon^4+4\varepsilon^3-6\varepsilon^2+4\varepsilon-1)+\delta^5\varepsilon^4(2\varepsilon^5-8\varepsilon^4+12\varepsilon^3-4\varepsilon^2-6\varepsilon+4)-\delta^4\varepsilon^3(2\varepsilon^4+5\varepsilon^3-14\varepsilon^2+25\varepsilon-1)+\delta^3\varepsilon^2(8\varepsilon^4-18\varepsilon^3+20\varepsilon^2-18\varepsilon+8)+\delta^2\varepsilon(8\varepsilon^3-20\varepsilon^2+4\varepsilon+8)+40\delta\varepsilon^2-40\delta\varepsilon+64\right)}{(\delta^3\varepsilon^7-2\delta^3\varepsilon^6+2\delta^3\varepsilon^5-2\delta^3\varepsilon^4+\delta^3\varepsilon^3+2\delta^2\varepsilon^3-2\delta^2\varepsilon^2+4\delta\varepsilon^4-4\delta\varepsilon^3+4\delta\varepsilon^2-4\delta\varepsilon+8)^2}$$

the BCJR algorithm, using the received sequence $\boldsymbol{y}$ and the incoming messages $L^{(\ell-1)}_{\mathrm{v}\to\mathcal{H},t}$ from the previous iteration of the channel decoder, where $L^{(0)}_{\mathrm{v}\to\mathcal{H},t}=0$. Then I_c decoding iterations are performed between the variable nodes and check nodes of the SC-LDPC code. At the variable node updates in decoding iteration i, the messages $L^{(\ell)}_{\mathcal{H}\to\mathrm{v},t}$ from the channel are combined with the incoming messages from the check nodes $L^{(i-1)}_{\mathrm{c}\to\mathrm{v},t}$ to produce the outgoing messages $L^{(i)}_{\mathrm{v}\to\mathrm{c},t}$. These are then used at the check node updates to produce the messages $L^{(i)}_{\mathrm{c}\to\mathrm{v},t}$. Using $I_\mathrm{c}>1$ decoding iterations between the channel detector updates reduces the overall complexity of the receiver.

4.1 Density Evolution for ISI Channels with Erasures

In case of ISI channels with erasures, all L-values exchanged in the message passing receiver can take the values $L=-\infty$ or $L=+\infty$ if the corresponding variable is a known -1 or $+1$, repsectively, or $L=0$ if the variable is an erasure. For this reason, the messages -1, $+1$ and '?' can be exchanged instead of L-values, and density evolution is equivalent to tracking the evolution of the erasure probabilities of the variables.

The probability $q^{(\ell)}_{\mathcal{H},t}$ that an outgoing message $L^{(\ell)}_{\mathcal{H}\to\mathrm{v},t}$ of the channel detector is an erasure is given by the transfer function derived in Section 3, i.e.,

$$q^{(\ell)}_{\mathcal{H},t}=g\left(\delta_t^{(\ell-1)},\varepsilon\right)\ ,$$

where $\delta_t^{(\ell-1)}$ denotes the erasure probability of the incoming messages $L^{(\ell-1)}_{\mathcal{H}\to\mathrm{v},t}$ from the previous code-channel iteration.

For the decoding of the SC-LDPC code, the density evolution equations can be derived fololwing the approach in [11]. At a variable node update, the average probability that a message $L^{(i)}_{\mathrm{v}\to\mathrm{c},t}$ is an erasure is given by

$$p_t^{(i)}=q^{(\ell)}_{\mathcal{H},t}\cdot\frac{1}{m+1}\sum_{j=0}^{m}\left(q^{(i-1)}_{\mathrm{c},t+j}\right)^{d_v-1}\ , \tag{5}$$

where

$$q^{(i)}_{\mathrm{c},t}=\frac{1}{m+1}\sum_{j=0}^{m}\left(1-\left(1-p^{(i)}_{t-j}\right)^{d_c-1}\right) \tag{6}$$

denotes the average erasure probability of the messages $L^{(i)}_{\mathrm{c}\to\mathrm{v},t}$ computed at a check node update. Equations (5) and (6) are valid for a (d_v,d_c)-regular LDPC code that is spatially coupled in such a way that edges are spread uniformly over $m+1$ spatial positions.

A message $L^{(\ell)}_{\mathrm{v}\to\mathcal{H},t}$ passed to the channel detector after I_c decoding iterations is erased if all d_v messages $L^{(I_\mathrm{c})}_{\mathrm{c}\to\mathrm{v},t}$ are erased. The corresponding erasure probability is hence equal to

$$\delta_t^{(\ell)}=\frac{1}{m+1}\sum_{j=0}^{m}\left(q^{(i-1)}_{\mathrm{c},t+j}\right)^{d_v}\ .$$

Table 4: Thresholds of (3,6) LDPC code for ISI channels with erasures and AWGN

	CH-I		CH-II		CH-III	
	Erasures	AWGN	Erasures	AWGN	Erasures	AWGN
$\varepsilon^{\mathrm{BP}}, \gamma^{\mathrm{BP}}$	0.5689	1.703	0.7055	2.598	0.8254	5.474
$\varepsilon^{\mathrm{MAP}}, \gamma^{\mathrm{MAP}}$	0.6387	1.160	0.7519	1.509	0.8482	2.975
h^{BP}	0.8530	0.8510	1.5870	1.5147	3.3010	2.9177
$\mathrm{h}^{\max}$	1.500	1.500	2.250	2.250	4.000	4.000
$\mathrm{h}^{\mathrm{BP}}/\mathrm{h}^{\max}$	0.5689	0.5673	0.7055	0.6732	0.8254	0.7294
$\mathrm{h}^{\mathrm{MAP}}$	0.9580	0.9195	1.6918	1.6200	3.3926	3.2146
$\mathrm{h}^{\mathrm{MAP}}/\mathrm{h}^{\max}$	0.6387	0.6130	0.7519	0.7197	0.8482	0.8036
$\varepsilon^{\mathrm{ISR}}, \gamma^{\mathrm{ISR}}$	0.6404	0.823	0.7530	1.437	0.8506	2.960

The average GEXIT function can be expressed as [12]

$$\frac{1}{n}\frac{dH(\mathcal{X}|\mathcal{Y}, S_0)}{d\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} H(Z_i|\mathcal{Y}_{\sim i}, S_0) \ ,$$

where the channel parameter is taken to be $\mathrm{h}_i = \varepsilon$.

From the GEXIT function we can thus define the generalized BP EXIT (GB-EXIT) function, h_i^{BP}, as the joint iterative decoding of Z_i from $\mathcal{Y}_{\sim i}$ and S_0. The exact GB-EXIT function is computed in the same fashion as in [12]. That is,

$$\mathrm{h}^{\mathrm{BP}} = \sum_{i,j} P(\boldsymbol{m}_\alpha^{(i)}) H(Z_i|\alpha_t = \boldsymbol{m}_\alpha^{(i)}, \beta_{t+1} = \boldsymbol{m}_\beta^{(j)}) P(\boldsymbol{m}_\beta^{(j)}) \ .$$

The GB-EXIT function is used as described in [12] to compute an upper bound $\varepsilon^{\mathrm{MAP}}$ on the MAP threshold. The SIR also can be computed using the state distribution of the Markov chain from Section 3. Using the notation $\mathcal{X}$ for the sequence $X_1 X_2 \ldots X_t$ the SIR is computed using the expression $I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X})$, with $H(\mathcal{Y}|\mathcal{X}) = h_b(\varepsilon)$, where $h_b(\cdot)$ denotes the binary entropy function. The entropy rate $H(\mathcal{Y})$ can be computed by using the definition [9]

$$H(\mathcal{Y}) = \lim_{t\to\infty} H(Y_t|Y^{t-1}) = \sum_{i=1}^{|\mathcal{M}_\alpha|} P_{\alpha_t}(\boldsymbol{m}_\alpha^{(i)}) H(Y|\alpha_t = \boldsymbol{m}_\alpha^{(i)}).$$

$P_{\alpha_t}(\boldsymbol{m}_\alpha^{(i)})$ is the i-th entry in the steady state distribution $\boldsymbol{\pi}_{\alpha 1}$ computed without using *a-priori* inputs from the code. This can be obtained from the steady state distribution $\boldsymbol{\pi}_\alpha(\varepsilon, \delta)$ by setting the erasure probability from the code to 1, i.e., $\boldsymbol{\pi}_{\alpha 1}(\varepsilon) = \boldsymbol{\pi}_\alpha(\varepsilon, 1)$.

4.2 Density Evolution for ISI Channels with AWGN

For channels with AWGN, no explicit transfer functions are available for computing the probability densities $p(L^{(\ell)}_{\mathcal{H}\to \mathrm{v},t})$ of the outgoing messages at the channel detector. For this reason we use Monte Carlo simulations for the BCJR detector to determine these densities from the incoming densities $p(L^{(\ell)}_{\mathrm{v}\to\mathcal{H},t})$ and the noise distribution.

Table 5: Thresholds of regular codes with spatial coupling for ISI channels with erasures and AWGN

		Erasures						AWGN					
Code	Channel	$\varepsilon^{\mathrm{BP}}$	ε^{1}	ε^{3}	ε^{6}	$\varepsilon^{\mathrm{MAP}}$	$\varepsilon^{\mathrm{ISR}}$	γ^{BP}	γ^{1}	γ^{3}	γ^{10}	γ^{MAP}	γ^{ISR}
$(3,6)$	CH-I	0.5689	0.6386	0.6386	0.6386	0.6387	0.6404	1.703	1.330	1.240	1.178	1.160	0.823
	CH-II	0.7055	0.7519	0.7519	0.7519	0.7519	0.7530	2.598	1.598	1.587	1.535	1.509	1.437
	CH-III	0.8254	0.8482	0.8482	0.8482	0.8482	0.8506	5.474	3.019	3.010	2.998	2.975	2.960
$(4,8)$	CH-I	0.5100	0.6399	0.6401	0.6401	0.6404	0.6404	2.441	0.896	0.877	0.866	0.853	0.823
	CH-II	0.6618	0.7528	0.7528	0.7528	0.7530	0.7530	3.596	1.494	1.478	1.458	1.448	1.437
	CH-III	0.7997	0.8501	0.8501	0.8501	0.8501	0.8506	6.745	3.100	3.061	2.993	2.963	2.960
$(5,10)$	CH-I	0.4647	0.6400	0.6403	0.6403	0.6404	0.6404	3.029	0.877	0.852	0.844	0.834	0.823
	CH-II	0.6275	0.7526	0.7529	0.7529	0.7530	0.7530	4.348	1.483	1.461	1.450	1.437	1.437
	CH-III	0.7775	0.8503	0.8503	0.8503	0.8503	0.8506	7.550	3.036	3.000	2.987	2.960	2.960
$(6,12)$	CH-I	0.4647	0.6378	0.6403	0.6403	0.6404	0.6404	3.517	0.853	0.844	0.829	0.823	0.823
	CH-II	0.6000	0.7504	0.7529	0.7530	0.7530	0.7530	4.938	1.483	1.461	1.450	1.437	1.437
	CH-III	0.7588	0.8504	0.8504	0.8504	0.8504	0.8506	8.123	3.036	2.990	2.980	2.960	2.960

For the message passing decoding of the SC-LDPC code we use discretized density evolution [13], which is exact up to the numerical accuracy and the resolution of the underlying quantization. During the i-th iteration within the code, at the variable nodes the message densities $p(L^{(i)}_{\mathrm{v}\to\mathrm{c},t})$ along an edge can be obtained as the convolution of the density $p(L^{(\ell)}_{\mathcal{H}\to\mathrm{v},t})$ from the channel with the $d_v - 1$ densities $p(L^{(i)}_{\mathrm{c}\to\mathrm{v},t})$ of incoming messages from the other edges. This can be implemented efficiently using the FFT. At the check nodes, the densities $p(L^{(i)}_{\mathrm{c}\to\mathrm{v},t})$ are computed from the $d_c - 1$ incoming densities $p(L^{(i)}_{\mathrm{v}\to\mathrm{c},t})$ in a nested fashion from a two-dimensional lookup table [13]. After the I_{c} iterations between check and variable nodes are completed, the density $p(L^{(\ell)}_{\mathrm{v}\to\mathcal{H},t})$ of messages passed to the channel is obtained as the convolution of all d_v message densities $p(L^{(I_{\mathrm{c}})}_{\mathrm{c}\to\mathrm{v},t})$.

The GEXIT function for the BIAWGNC with ISI can be defined as [8]

$$G(\mathrm{h}) = \frac{1}{n}\frac{dH(\mathcal{X}|\mathcal{Y}, S_0)}{d\mathrm{h}}$$

where

$$\mathrm{h} = H(Z_i|Y_i) = H(Z_i) - I(Y_i; Z_i), \tag{7}$$

with

$$I(Y_i; Z_i) = \int_{-\infty}^{\infty} \sum_{z_i} P(y_i, z_i) \log_2 \left\{ \frac{P(y_i|z_i)}{\sum_{\boldsymbol{z}'_i} P(z'_i) P(y_i|z'_i)} \right\} dy.$$

The entropy h is a function of the channel parameter ε which is chosen for convenience to be $\varepsilon = -\frac{1}{2\sigma^2}$. The signal-to-noise ratio parameter used in this paper, γ is the ratio of bit energy, E_b, to the noise spectrum density, N_o, in decibels. Using the GBP-EXIT function, an upper bound γ^{MAP} on the MAP threshold is computed as described in [8]. The SIR is computed numerically using the method described in [14].

5 Numerical Results

If we compare the BP thresholds of the channels by looking at the channel erasure probability ε in Table 4 for the regular (3,6) code we observe that CH-III

has the best performance (the threshold occurs at the highest erasure probability) followed by CH-II while CH-I has the worst performance. On the other hand, if we look at the Gaussian channel we notice that the ordering is reversed with CH-I having the best threshold (at lowest SNR) and CH-III has the worst performance. Thus we see an apparent inconsistency in the performance of the channels when erasures are changed to AWGN. This is also true for the MAP threshold and the ISR.

In Table 5 we see the same phenomenon when different codes are used. That is, the channel ranking for the BP and MAP thresholds as well as the ISR is reversed for all codes when changing from erasures to AWGN.

But we can also characterize the thresholds in terms of the entropy $\mathrm{h} = H(Z_i|Y_i)$. This is defined in (7) for the Gaussian channel, while for the erasure channel we observe that

$$H(Z_i|Y_i) = \bar{\varepsilon}H(Z_i|Y_i \neq ?) + \varepsilon H(Z_i|Y_i = ?) = \varepsilon H(Z_i),$$

since Z_i is known with certainty when Y_i is not erased and it is independent of it when it is erased.

Representing the thresholds in terms of entropy, we now observe in Table 4 that the ranking of the channels is unchanged when changing between erasures and AWGN. I.e., CH-III has the best performance with both AWGN and erasures by having thresholds which are at the highest entropy, while CH-I has the worst performance. This could be attributed to the fact the CH-III has the highest $H(Z_i)$ (labelled $\mathrm{h}^{\max}$ in the table). But if we normalize the threshold by dividing by $\mathrm{h}^{\max}$ the ranking is unchanged. It is interesting to note that the BP entropies h^{BP} for erasures and AWGN are relatively close to each other for a given ISI channel. On the other hand, their gap is still to large for making an accurate prediction from the erasure to the Gaussian case.

In Table 5 we can also see that with uncoupled regular codes the MAP threshold improves with increasing variable node degree, while the BP threshold becomes worse. With spatially-coupled codes, we observe that the BP thresholds approach the MAP thresholds of the uncoupled codes for all three channels. It is also interesting to see that for the (5,10) and (6,12) code the MAP threshold is equal or very close to the ISR for all three channels. This demonstrates that with spatial coupling we can universally approach the ISR of different ISI channels using a single code. This makes spatially coupled codes superior to uncoupled irregular codes that need to be optimized for a particular ISI channel, which cannot guarantee robust performance if the channel is changed.

6 Conclusions

We have derived exact transfer function for three different channels using a method which can be applied to any arbitrary ISI channel. We have further shown that to compare the behaviour of ISI channels with erasures and AWGN, the proper parameter is the entropy $H(Z_i|Y_i)$, by which we can observe a consistent behaviour. Finally we have shown numerically that with spatially coupled

LDPC codes we can universally achieve the SIR of different ISI channels.

References

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

[2] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

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

[4] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, 2017.

[5] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Allerton*, Monticello, IL, USA, Sept./Oct. 2010, pp. 347–353.

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

[7] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[8] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially coupled codes on intersymbol interference channels," in *Proc. IEEE Int Conf. on Communications (ICC)*, Ottawa, ON, Canada, June 2012, pp. 2181–2186.

[9] H. D. Pfister and P. H. Siegel, "On the capacity of finite state channels and the analysis of convolutional accumulate-m codes," Ph.D dissertation, Dept. Elect. Eng., UC San Diego, USA, 2003.

[10] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel," in *Proc. GLOBECOM* , San Francisco, CA, USA, Dec. 2003, pp. 1741–1745.

[11] M. Lentmaier, M. B. S. Tavares, and G. P. Fettweis, "Exact erasure channel density evolution for protograph-based generalized LDPC codes," in *Proc. Int. Symp. Inf. Theory*, Seoul, South Korea, June/July 2009, pp. 566–570.

[12] C. Wang and H. D. Pfister, "Upper bounds on the MAP threshold of iterative decoding systems with erasure noise," in *Proc. . Int. Symp. Turbo Codes & Related Topics*, Lausanne, Switzerland, Sept. 2008, pp. 7–12.

[13] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, 2001.

[14] D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug 2006.

Paper III

Robust Performance Over Changing Intersymbol Interference Channels by Spatial Coupling

We show that spatially coupled low-density parity-check (LDPC) codes yield robust performance over changing intersymbol interfere (ISI) channels with optimal and suboptimal detectors. We compare the performance with classical LDPC code design which involves optimizing the degree distribution for a given (known) channel. We demonstrate that these classical schemes, despite working very good when designed for a given channel, can perform poorly if the channel is exchanged. With spatially coupled LDPC codes, however, we get performances close to the symmetric information rates with just a single code, without the need to know the channel and adapt to it at the transmitter. We also investigate threshold saturation with the linear minimum mean square error (LMMSE) detector and show that with spatial coupling its performance can get remarkably close to that of an optimal detector for regular LDPC codes.

Keywords: Spatial Coupling, Inter-Symbol interference, Equalization, Threshold Saturation, Universality, Robust performance, Degree optimization

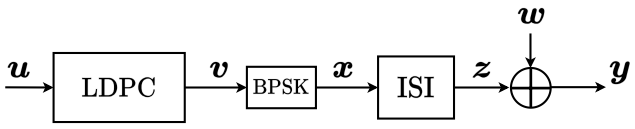


Figure 1: Block diagram showing the transmitter and the ISI channel.

1 Introduction

Spatially coupled codes were first studied in [1, 2] in the context of low-density parity-check (LDPC) codes and later applied to other classes of codes [3]. They are known to exhibit remarkably good performance in a range of coding scenarios [4–6] and scenarios beyond coding as well [7, 8]. This good performance is a consequence of the fact that the belief propagation (BP) threshold of the coupled code approaches the threshold of the underlying uncoupled code with maximum a-priori (MAP) decoding, a phenomenon known as threshold saturation. This phenomenon was proved in [9,10] for binary memoryless symmetric (BMS) channels. For such channels, it has been shown that spatially coupled LDPC codes can universally achieve capacity with BP decoding [10].

In [11], it was shown that threshold saturation also occurs for channels with memory. The authors also showed that, with regular codes of high node degree, the BP threshold of the coupled code approaches the symmetric information rate (SIR) of the simple dicode channel. In [12], the same phenomenon was demonstrated for three different intersymbol interference (ISI) channels with larger memory, and it was shown that a single code universally achieves the SIR of the three considered ISI channels. It can be observed that spatial coupling opens up a new paradigm of code design whereby the global MAP threshold, which was considered practically unattainable, now matters and can be achieved with the locally optimal BP decoding. One may then choose a code with good MAP threshold (which often implies bad BP threshold) and apply spatial coupling to attain the MAP threshold with BP decoding.

In this paper, we apply spatially coupled LDPC codes to turbo equalization demonstrating that their universality provides practical advantages when compared to classical code design. Classical code design for turbo equalization usually involves optimizing a code for a particular channel. For LDPC codes, this often means optimizing the degree distribution for the considered channel [13–15]. The shortcoming is that a degree distribution optimized for a given channel may perform poorly over a different channel. Furthermore, one needs to know the channel at the transmitter. In contrast, due to their universal behavior, no optimization is required for spatially coupled LDPC codes, hence there is no need to know the channel and they are expected to provide superior performance for scenarios where the channel changes. We compare the robustness of regular spatially coupled LDPC codes across three different ISI channels against optimized irregular LDPC codes in terms of thresholds and finite length simulations. We also show that with regular codes the suboptimal linear minimum mean square error (LMMSE) detector, which has a very poor performance when compared to the optimal detector, has a performnce quite close to the BCJR detector when spatially coupled LDPC codes are used with both of them.

The remainder of the paper is organized as follows. After introducing the

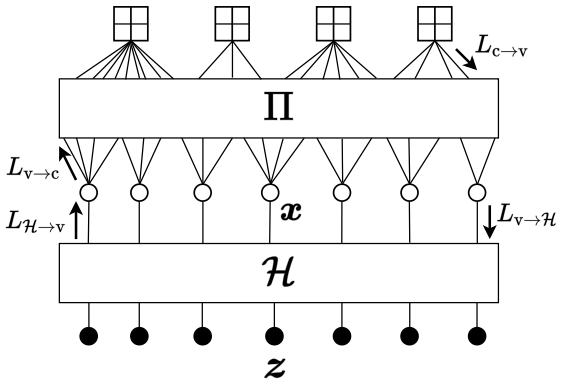


Figure 2: Factor graph of turbo equalization with an irregular LDPC code.

considered system model in Section 2, we first discuss classical code design for ISI channels with irregular LDPC codes to highlight the weakness of such approach in Section 3. In Section 4, we describe code design with spatially coupled LDPC codes for an optimal detector and how their universality can overcome these problems. In Section 5, we consider a suboptimal LMMSE detector, discuss threshold saturation in this setting and show that we cannot simply use the area bound to provide a meaningful upper bound for the coupled threshold for such a detector. We then propose a method to approximate such threshold. Finally, the paper is concluded in Section 6.

2 System Model

The system model is shown in Fig. 1. A sequence of k information bits $\boldsymbol{u}$ is encoded onto a codeword $\boldsymbol{v}$ of length n. The codeword is then mapped into a sequence of symbols $\boldsymbol{x}$ using binary phase shift keying (BPSK) modulation with the mapping $0 \mapsto +1$ and $1 \mapsto -1$. The sequence $\boldsymbol{x}$ is transmitted over an ISI channel of memory ν. The output of the channel filter, $\boldsymbol{z}$, is the convolution of $\boldsymbol{x}$ and the impulse response of the channel, $\boldsymbol{h}$, which has $\nu+1$ taps. Table 1 shows the impulse responses of the considered ISI channels. The impulse responses are normalized such that $||\boldsymbol{h}|| = 1$. This makes the signal energy at the receiver, E_z, equal to the signal energy, E_x, at the input of the ISI channel. The received sequence, $\boldsymbol{y}$, is the result of corrupting $\boldsymbol{z}$ by additive white Gaussian noise. We define $\gamma = E_\mathrm{b}/N_0$, where E_b is the average energy per information bit and $N_0 = 2\sigma^2$ the noise spectrum density. Note that $E_\mathrm{b}/N_0 = E_z/RN_0$, where R is the rate of the code.

At the receiver, the channel detector and the decoder exchange information iteratively, a process widely known as turbo equalization. We use BP decoding to decode the LDPC code while for the channel we consider two types of detectors—the optimal detector, implemented by applying the BCJR algorithm, and the suboptimal LMMSE detector. The detectors are implemented as described in [16]. We thus have message exchanges within the decoder, i.e., between check nodes (CNs) and variable nodes (VNs) for I_C iterations, and message exchanges between the VNs and the detector for I_D iterations, as depicted by Fig. 2. The messages exchanged are log-likelihood ratios.

3 Code Design for Turbo Equalization with Irregular LDPC Codes

We consider irregular LDPC codes with polynomials $\lambda(x)$ and $\mathsf{L}(x)$ representing the VN degree distribution from an edge and node perspective, respectively, while the CN degree distribution from an edge perspective is represented by $\rho(x)$. The design rate for such a code is given by $R(\lambda,\rho) = 1 - \frac{\int_0^1 \rho(x)}{\int_0^1 \lambda(x)}$. We consider a design rate of 1/2 for all scenarios. To achieve good performance with turbo equalization with classical code design, the LDPC code needs to be specifically designed for a given channel. One approach is to optimize the degree distribution such that the BP threshold is maximized. This can be achieved via density evolution. We briefly describe density evolution for irregular LDPC codes over an ISI channel.

The density passed from the detector to the code, $p(L^{(\ell)}_{\mathcal{H}\to \mathrm{v}})$, is a function of the incoming densities from all VNs $p(L^{(\ell-1)}_{\mathrm{v}\to\mathcal{H}})$ and the noise distribution. Denoting this function as $\mathcal{T}$ we have

$$p(L^{(\ell)}_{\mathcal{H}\to \mathrm{v}}) = \mathcal{T}\left(p(L^{(\ell-1)}_{\mathrm{v}\to\mathcal{H}}), \sigma\right) .$$

It is not possible to obtain $\mathcal{T}(.,.)$ in closed form for Gaussian noise (it can be computed for erasure noise [12]), but it can be evaluated via Monte Carlo methods. For each VN, its outgoing density to an edge is the convolution of the density $p(L^{(\ell)}_{\mathcal{H}\to \mathrm{v}})$ from the detector and the $d_{\mathsf{v}} - 1$ incoming densities from other edges, where d_{v} is the VN degree. The average density from the VNs to a neighboring CN, $p(L^{(i)}_{\mathrm{v}\to \mathrm{c}})$, is then obtained by averaging over the degree distribution $\lambda(x)$. At each CN of degree d_{c}, the outgoing density is computed from the $d_{\mathsf{c}} - 1$ incoming densities in a nested fashion using a two-dimensional lookup table for discretized density evolution [17]. Similar to the VNs, the average density to a VN, $p(L^{(i)}_{\mathrm{c}\to \mathrm{v}})$, is obtained by averaging over the degree distribution $\rho(x)$. After I_{C} iterations within the code, the density passed from a VN to the detector, $p(L^{(\ell)}_{\mathrm{v}\to\mathcal{H}})$, is the convolution of the incoming d_{v} densities from the neighboring CNs. The density evolution update equation for the joint BP decoding of the code and channel is thus given as

$$\begin{aligned}
p(L^{(i)}_{\mathrm{v}\to \mathrm{c}}) &= p(L^{(\ell-1)}_{\mathcal{H}\to \mathrm{v}}) \circledast \boldsymbol{\lambda}\left(p(L^{(i-1)}_{\mathrm{c}\to \mathrm{v}})\right) \\
p(L^{(\ell)}_{\mathrm{c}\to \mathrm{v}}) &= \boldsymbol{\rho}\left(p(L^{(i-1)}_{\mathrm{v}\to \mathrm{c}})\right) \\
p(L^{(\ell)}_{\mathrm{v}\to\mathcal{H}}) &= \mathsf{L}\left(L^{(I_{\mathrm{C}})}_{\mathrm{c}\to \mathrm{v}})\right) \\
p(L^{(\ell)}_{\mathcal{H}\to \mathrm{v}}) &= \mathcal{T}\left(p(L^{(\ell)}_{\mathrm{v}\to\mathcal{H}})\right)
\end{aligned}$$

where for a given density a, $\boldsymbol{\lambda}(a) = \sum_j \lambda_j a^{\circledast(j-1)}$, $\boldsymbol{\rho}(a) = \sum_j \rho_j a^{\boxed{*}(j-1)}$ and $\mathsf{L}(a) = \sum_i \mathsf{L}_i a^{\circledast(j)}$. The operator $\circledast$ represents the convolution of densities while $\boxed{*}$ represents the density transformations at the CN as used in [18].

To find a code for a particular channel, we use a two-step searching scheme. In the first step, a list of codes is generated using the EXIT chart design method. This is done by combining the code's VNs with the detector and fitting the EXIT

Table 1: Discrete impulse responses of the considered ISI channels

CH-I	$\boldsymbol{h} = [\ 0.7071 \quad -0.7071\ ]$	$\nu = 1$
CH-II	$\boldsymbol{h} = [\ 0.408 \quad 0.816 \quad 0.408\ ]$	$\nu = 2$
CH-III	$\boldsymbol{h} = [\ 0.227 \quad 0.46 \quad 0.688 \quad 0.46 \quad 0.227\ ]$	$\nu = 4$

Table 2: Codes optimized for the BCJR detector

CH-I			CH-II			CH-III		
i	λ_i	ρ_i	i	λ_i	ρ_i	i	λ_i	ρ_i
2	0.3075		2	0.3963		2	0.5935	
3	0.3208		3	0.3589		3	0.0243	
5	0.0180	0.0436	4		0.0458	6		0.8856
6	0.0377		5	0.0153	0.0189	7		0.1144
7		0.9456	6		0.9128	11	0.0182	
8		0.0108	8		0.0225	17	0.3639	
10	0.0130		9	0.0592				
13	0.0446		13	0.1583				
16	0.0876		17	0.0120				
18	0.1708							

curve of the combined node with that of the CNs. The method is used according to the description in [13] but with the modification that we use CNs with more than one degree. The fitting is done manually at an SNR close to the SIR of the channel. The SIR values were calculated using a numerical method described in [19]. Due to the approximate nature of the EXIT chart approach, a list of N_{Z} codes whose curves closely fit (imperfections are allowed where we might have a small crossing of the curves) around the SIR rate are generated. In the second step, we perform density evolution for each of the N_{Z} codes and select the code with the best BP threshold. Table 2 shows the codes obtained for the BCJR detector while those for the LMMSE detector are shown in Table 3. With this procedure, we can find codes with threshold close to the SIR of the particular channel the code is designed for. This approach, however, does not guarantee that a code designed for a specific channel performs well for other ISI channels or a time-varying channel. Tables 4 and 5 show the BP thresholds of the designed codes for each of the three considered channels[1]. In the tables each bold entry represents the threshold of a code matched to the channel it was designed for. We observe that when a code designed for a given ISI channel is applied to a different channel, the gap to the SIR can be large. For example, the code designed for CH-II under the LMMSE detector has a threshold 0.1 dB away from the corresponding SIR, but the code threshold of the same code is nearly 3 dB away for channel CH-III.

4 Code Design with Spatially Coupled Codes

We now consider spatially coupled LDPC codes with a BCJR detector. The whole system can be described by a factor graph which combines the graph

[1]With a slight abuse of notation, we use the term *BP thresholds* for the system with LMMSE detector as well, even though it is not locally optimal as the BCJR detector.

Table 3: Codes optimized for the LMMSE detector

CH-I			CH-II			CH-III		
i	λ_i	ρ_i	i	λ_i	ρ_i	i	λ_i	ρ_i
2	0.2652		2	0.3131		2	0.4792	0.0270
3	0.2921		3	0.2805		3	0.0357	
4	0.0489	0.0270	4	0.0123	0.0496	4	0.0172	0.0696
5		0.0663	6		0.1662	5	0.0209	0.0150
6	0.0178		8		0.7841	7		0.0632
8		0.9067	10	0.0139		8		0.8252
12	0.0306		16	0.2225		19	0.4348	
14	0.0681		20	0.1578		50	0.0122	
18	0.0419							
20	0.2355							

Table 4: Code design and changing channel with BCJR

Designed for	BP threshold when applied to CH-I	CH-II	CH-III	SIR
CH-I	**0.93**	1.65	4.55	0.82
CH-II	1.42	**1.51**	4.11	1.44
CH-III	3.29	3.32	**3.25**	2.96

representing the channel constraints and the Tanner graph of the LDPC code. Specifically, the factor graph is constructed by placing L copies of a $(d_{\mathsf{v}}, d_{\mathsf{c}})$ regular LDPC code of VN degree d_{v} and CN degree d_{c} in L spatial positions in the range $\mathcal{L} \in \{1, \ldots, L\}$. Fig. 3 shows the factor graph for three spatial positions, $t-1$, t, and $t+1$. Each spatial position consists of N VNs, represented by empty circles, and M CNs ($M = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} N$), represented by squares with a cross. The L copies are coupled as follows: each VN at position $t \in \mathcal{L}$ is connected to CNs in the range $[t, \ldots, t+m]$, where m is referred to as the coupling memory. Hence, each CN at position t is connected to VNs in the range $[t-m, \ldots, t]$. The constraints of the ISI channel at each spatial position are represented by a square labeled with the letter $\mathcal{H}$, referred to as factor node. Each of the VNs represented by the black circles at the bottom of the figure (denoted by $\{\boldsymbol{z}_t\}$) represent a block of N symbols at the output of the ISI channel before being corrupted by noise. This means that $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L)$ (see Fig. 1). The rectangles at each spatial position between the Tanner graph of the SC-LDPC code and the channel factor nodes represent multiplexers that multiplex the N code bits at each spatial position into a single binary sequence ($\boldsymbol{x}_t$, with $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$) at the input of the channel. This makes $n = NL$. Decoding is then performed by iteratively passing messages along the edges of the graph in Fig. 3.

Owing to the universality of spatially coupled LDPC codes, we perform code design for ISI channels in two steps. We first pick a regular code with high node degree and then apply spatial coupling to achieve good performance over all channels. We apply this principle and consider the $(5, 10)$ and $(6, 12)$ regular LDPC codes. In Table 6 we give the thresholds for the uncoupled codes (labeled γ^{BP}) and those of the coupled codes for different coupling memories (labelled

Table 5: Code design and changing channel with LMMSE

Designed for	BP threshold when applied to CH-I	CH-II	CH-III	SIR
CH-I	**1.02**	1.96	6.53	0.82
CH-II	1.37	**1.54**	5.80	1.44
CH-III	3.52	3.44	**3.35**	2.96

$\gamma^{(m)}$). We also show the corresponding MAP thresholds and the SIRs as well. It is observed that uncoupled regular LDPC codes have poor BP threshold, with a gap up to 5 dB from the SIR. The MAP thresholds, however, are almost equal to the SIRs[2]. Due to threshold saturation, the BP threshold of spatially coupled LDPC codes are close to the SIR for the three different ISI channels. For the $(6, 12)$ code, the threshold for memory $m = 6$ is good for all channels and better than the ones of the optimized codes in Table 4.

In Fig. 4, we give simulation results for the $(6, 12)$ code for CH-II and CH-III and compare it to the irregular LDPC code designed for CH-II. For the spatially coupled LDPC code, we use $m = 6$ and $L = 500$ with $N = 10\,000$, which gives the design rate $R = 0.494$. The code is decoded using window decoding [20] with a window size of $W = 30$, resulting in a decoding latency of $WN = 300\,000$ symbols. Within the window, we use $I_{\mathrm{C}} = 30$ iterations in the code and $I_{\mathrm{D}} = 20$ iterations between the code and the channel. For the irregular code, we use a block length of $n = 300\,000$ and the parity-check matrix is generated by the progressive edge growth algorithm [21]. We use $I_{\mathrm{C}} = 30$ iterations within the code and $I_{\mathrm{D}} = 20$ iterations between the code and the channel. The results indicate a convergence to the thresholds in Tables 4 and 6, whereby the gaps are smaller for the uncoupled code. The irregular code designed for CH-II performs very well for that particular channel, but the performance deteriorates significantly when the channel is changed to CH-III. The coupled code shows good performance for both channels.

5 Spatially Coupled Codes with the LMMSE Detector

In this section, we consider the use of spatially-coupled LDPC codes with a suboptimal channel detector, namely the LMMSE detector. For the LMMSE detector, we also observe in Table 6 that the BP threshold of spatially coupled LDPC codes improves with increasing memory. It is not clear, however, which value the coupled threshold saturates to. This is because using the generalized extrinsic information transfer (GEXIT) bounding technique as for the BCJR detector will not give us a meaningful bound. This can be observed by noting that the upper bound to the MAP threshold is based on the following facts. The GEXIT of an ISI channel with entropy $\mathsf{h} = \mathrm{H}(Z|Y)$ and initial state S_0 is

[2]The MAP thresholds for the system with BCJR detector were obtained from [12] where they were obtained using the area theorem.

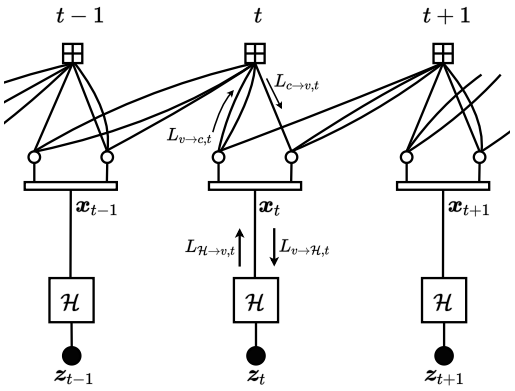


Figure 3: Compact graph representation for equalization with a (3,6) SC-LDPC code with coupling memory $m = 1$.

defined as [11]

$$\mathrm{G}(\mathsf{h}) = \frac{1}{n}\frac{\mathrm{dH}(\boldsymbol{X}|\boldsymbol{Y}, S_0)}{\mathrm{d}\mathsf{h}}.$$

The conditional entropy rate $\mathrm{H}(\boldsymbol{X}|\boldsymbol{Y}, S_0)$ can be thought of as the output of the globally optimal receiver (MAP receiver). But if we use a locally optimal receiver (using BP) instead, the output entropy will be greater than or equal to that of a globally optimal receiver. This is because the globally optimal receiver will reduce the uncertainty about $\boldsymbol{X}$ compared to the locally optimal receiver. By definition, the integral of the GEXIT equals the rate of the code as n grows to infinity and it is equal to zero below the MAP threshold. We thus have

$$R = \lim_{n\to\infty} \int_{\mathsf{h}^{\mathrm{MAP}}}^{1} \mathrm{G}(\mathsf{h}) \leq \int_{\mathsf{h}^{\mathrm{MAP}}}^{1} \mathrm{G}^{\mathrm{BP}}(\mathsf{h}),$$

where $\mathrm{G}^{\mathrm{BP}}(\mathsf{h})$ is obtained from a BP decoder with both the block length n and number of iterations I_{D} approaching infinity. This provides a way to compute an upper bound for the MAP threshold [22]. This is done by finding $\bar{\mathsf{h}}$ such that it is the largest positive number such that

$$\int_{\bar{\mathsf{h}}}^{1} \mathrm{G}^{\mathrm{BP}}(\mathsf{h}) = R.$$

This bound is tight for a receiver with the BCJR detector. If we change the detector to a suboptimal detector like the LMMSE, the message passing receiver is no longer locally optimal thus the performance of the whole sytem is degraded. As a result, the LMMSE detector will result into greater entropy than the BCJR detector. This implies that

$$\int_{\mathsf{h}^{\mathrm{MAP}}}^{1} \mathrm{G}^{\mathrm{BCJR}}(\mathsf{h}) \leq \int_{\mathsf{h}^{\mathrm{MAP}}}^{1} \mathrm{G}^{\mathrm{LMMSE}}(\mathsf{h}).$$

Thus if we were to apply the same bounding technique, we would have an upper bound higher than that of the receiver with the BCJR detector. But since we know that the performance will be degraded by using a suboptimal detector, this bound becomes meaningless.

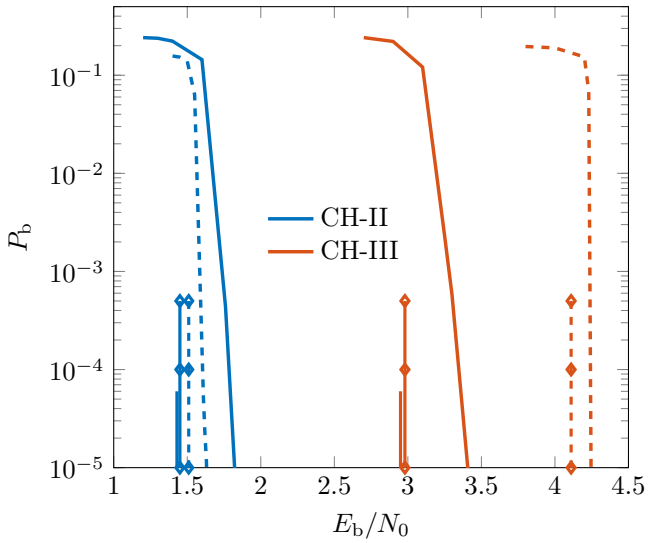


Figure 4: Simulation results with BCJR detector. Dashed lines show the performance of the irregular code optimized for CH-II while solid lines are for that of the (6,12) with $m = 6$. The solid short vertical lines are the corresponding SIRs while the long vertical lines with diamonds mark the BP thresholds.

Table 6: Thresholds of regular codes with spatial coupling for ISI channels with BCJR and LMMSE detectors

		BCJR						LMMSE					
Code	Channel	γ^{BP}	$\gamma^{(1)}$	$\gamma^{(6)}$	$\gamma^{(10)}$	γ^{MAP}	γ^{SIR}	γ^{BP}	$\gamma^{(1)}$	$\gamma^{(6)}$	$\gamma^{(10)}$	γ^{Area}	γ^{SIR}
$(5, 10)$	CH-I	3.03	0.88	0.85	0.84	0.83	0.82	3.55	0.98	0.97	0.97	1.08	0.82
	CH-II	4.35	1.483	1.45	1.45	1.44	1.44	5.84	1.71	1.68	1.68	1.82	1.44
	CH-III	7.55	3.04	2.99	2.99	2.96	2.96	13.31	3.62	3.59	3.58	3.71	2.96
$(6, 12)$	CH-I	3.52	0.85	0.84	0.83	0.82	0.82	4.18	0.98	0.96	0.96	1.12	0.82
	CH-II	4.94	1.48	1.45	1.45	1.44	1.44	7.00	1.71	1.68	1.67	1.85	1.44
	CH-III	8.12	3.04	2.98	2.98	2.96	2.96	14.15	3.69	3.59	3.56	3.72	2.96

We can however approximate this value using the positive gap condition for the EXIT curves [23]. This is done by combining the detector and the VNs of the LDPC code and computing the area between the EXIT curve of the combined node, h_{f}, and that of the CN, h_{g}. An estimate of the BP threshold of the coupled system is given by finding the SNR at which the area between the curves transitions from negative to positive as E_{b}/N_0 is increased. If we denote the function which gives the entropy of a symmetric Gaussian with mean μ by $\psi(\mu)$, the average output entropy from VNs to a CN, $\mathrm{h_{E,VN}}$, is given by

$$\mathrm{h_{E,VN}} = \sum_i \lambda_i \psi\Big((i-1)\psi^{-1}(\mathrm{h_{E,CN}}) + \psi^{-1}(\mathrm{h_{E,DET}})\Big) \tag{1}$$

where $\mathrm{h_{E,CN}}$ is the average entropy from a CN and $\mathrm{h_{E,DET}}$ is the output entropy from a detector. The output entropy of the detector is a function of the a-priori entropy to the detector $\mathrm{h_{A,DET}}$ and E_z/N_0,

$$\mathrm{h_{E,DET}} = \mathcal{D}\left(\mathrm{h_{A,DET}}, \frac{E_z}{N_0}\right).$$

The function $\mathcal{D}$ is computed by Monte Carlo simulations and for a given E_z/N_0 it can be approximated by a third order polynomial in the a-priori entropy,

$$\mathcal{D}\left(\mathrm{h_{A,DET}}, \frac{E_z}{N_0}\right) = c_3\mathrm{h}_{\mathrm{A,DET}}^3 + c_2\mathrm{h}_{\mathrm{A,DET}}^2 + c_1\mathrm{h_{A,DET}} + c_0\,, \tag{2}$$

where

$$\mathrm{h_{A,DET}} = \sum_i L_i \psi\Big(i\psi^{-1}(\mathrm{h_{E,CN}})\Big). \tag{3}$$

Substituting (3) and (2) into (1), we obtain the EXIT curve of the combined variable-detector node h_f for a given SNR.

For the CN, we use the dual approximation [18, p. 236] to obtain the output entropy as

$$\mathrm{h_{E,CN}} = 1 - \sum_i \rho_i \psi\Big((i-1)\psi^{-1}(1-\mathrm{h_{E,VN}})\Big)\,.$$

We use a numerical approximation for $\psi(\mu)$ and $\psi^{-1}(\mu)$ based on the numerical approximations given in [13] for the $J(\sigma)$ function which gives the mutual information of a symmetric Gaussian density. We use the fact that for a symmetric Gaussian density, $\mu = \frac{\sigma^2}{2}$ which implies $J(\sigma) = 1 - \psi(\frac{\sigma^2}{2})$. We thus have the recursion

$$h_\mathrm{g}(v) = 1 - \sum_i \rho_i \psi\Big((i-1)\psi^{-1}(1-v)\Big)$$

$$h_\mathrm{f}(u) = \sum_i \lambda_i \psi\Big((i-1)\psi^{-1}(u) + \psi^{-1}\big(\mathcal{D}_\mathrm{L}(u)\big)\Big)$$

where

$$\mathcal{D}_\mathrm{L}(u) = \mathcal{D}\Big(\sum_i L_i \psi\big(i\psi^{-1}(u)\big), \frac{E_z}{N_0}\Big).$$

Fig. 5 illustrates how this recursion can be used to approximate the coupled threshold for the LMMSE detector. In Table 6, we denote this approximate value by γ^{Area}. It can be seen that the coupled threshold for higher memory seems to saturate to a value close but not equal to γ^{Area}. It appears that γ^{Area} is a pessimistic estimate as the coupled threshold always exceeds it. On the other hand, γ^{Area} provides an efficient way to roughly predict the coupling gain directly from the uncoupled graph.

We can notice in Table 6 that the LMMSE detector without coupling performs quite bad when compared to optimized irregular codes in Table 3. With coupling, however, the thresholds are improved significantly. In contrast to the case with the BCJR detector in Section 4, the coupled codes with the LMMSE detector do not always outperform the optimized codes for the matched channels but they are nevertheless very close to them and we still have the robustness with changing channels.

We can also notice in Table 6 that the linear detector without coupling yields poor thresholds when compared to the optimal detector for the same code and channel, especially for channels with severe ISI (for example in CH-II and CH-III). The situation, however, changes significantly when spatial coupling is introduced. We see that the linear detector achieves performance very close to the optimal detector with spatial coupling. For CH-III, for example, the BP

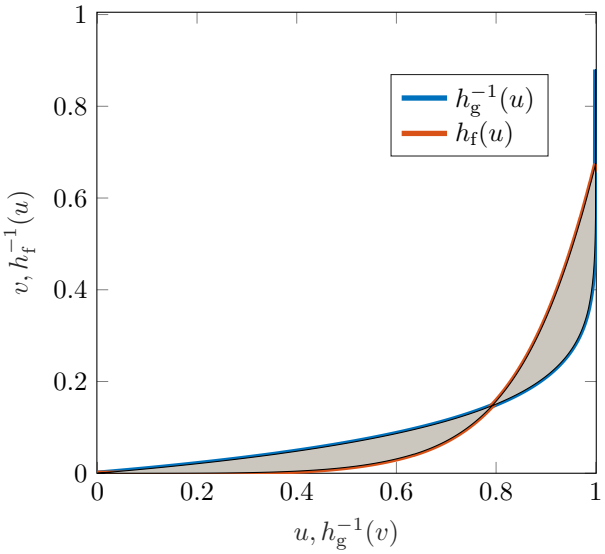


Figure 5: Approximating the coupled threshold of an (6,12) LDPC code and LMMSE equalizer for CH-II using the EXIT chart paradigm. The net area between the curve is zero at $\frac{E_{\mathrm{b}}}{N_0} = 1.85$ dB.

threshold of the $(5, 10)$ and $(6, 12)$ codes with LMMSE detector without coupling is more than 5 dB away from the threshold with the BJCR detector. With coupling, however, the gap is only a fraction of dBs for all coupling memories. In Fig. 6 this effect is shown with simulation results for the (6,12) code with the same settings as for the BCJR case discussed above for CH-III. In the figure we can clearly notice the narrowing of the gap between the two detectors when spatially coupled LDPC codes are applied.

6 Conclusions

We have demonstrated that spatially coupled LDPC codes are robust against both changing channel conditions and changing detector type when compared to classical code design. We can just use one LDPC code with high node degree with spatial coupling and attain universally good results for a number of ISI channels with BP decoding. It also makes the use of the suboptimal linear MMSE detector quite competitive with the BCJR detector. This can be of great practical value especially with situations in which applying the BCJR detector can be prohibitively complex due to, e.g., large channel memory.

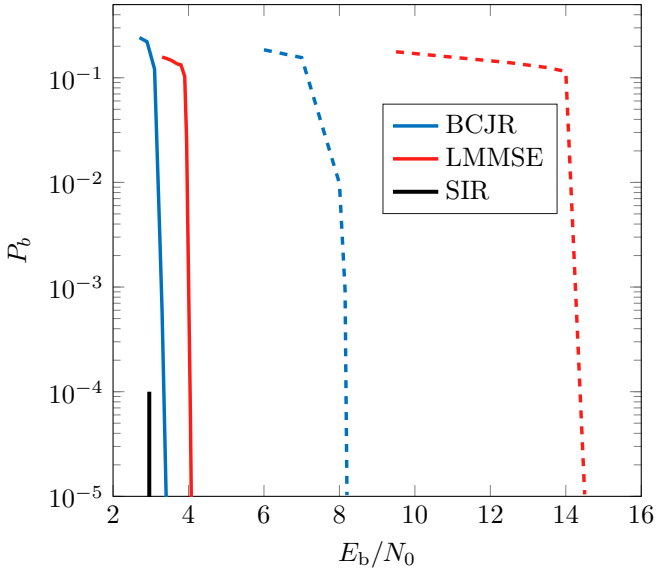


Figure 6: Simulation results comparing the BCJR vs LMMSE detector without coupling (dashed lines) and with coupling (solid lines) for a (6,12) LDPC code and CH-III.

References

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

[2] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[3] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, 2017.

[4] A. Yedla, P. S. Nguyen, H. D. Pfister, and K. R. Narayanan, "Universal codes for the gaussian MAC via spatial coupling," in *Proc. Allerton*, Monticello, IL, USA, Sept. 2011, pp. 1801–1808.

[5] F. Jardel, J. Boutros, V. Dedeoglu, M. Sarkiss, and G. Rekaya-Ben Othman, "Spatial coupling for distributed storage and diversity applications," in *Proc. COMNET*, Tunis, Tunisia, Nov. 2015, pp. 1–5.

[6] K.-H. Wang, W. Hou, S. Lu, P.-Y. Wu, Y.-L. Ueng, and J. Cheng, "Improving polar codes by spatial coupling," in *Proc. ISITA*, Singapore, Oct. 2018, pp. 432–436.

[7] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Allerton*, Monticello, IL, USA, Sept./Oct. 2010, pp. 347–353.

[8] V. Aref, N. Macris, and M. Vuffray, "Approaching the rate-distortion limit with spatial coupling, belief propagation, and decimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3954–3979, 2015.

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

[10] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.

[11] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *Proc. IEEE Int Conf. on Communications (ICC)*, Ottawa, ON, Canada, June 2012, pp. 2181–2186.

[12] M. M. Mashauri, A. G. i Amat, and M. Lentmaier, "On the universality of spatially coupled LDPC codes over intersymbol interference channels," in *Proc. ITW*, Kanazawa, Japan, Oct.2021.

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

[14] N. Varnica and A. Kavcic, "Optimized low-density parity-check codes for partial response channels," *IEEE Communications Letters*, vol. 7, no. 4, pp. 168–170, 2003.

[15] L. Kong, G. Han, Y. L. Guan, K. Cai, and K.-S. Chan, "Exit chart based LDPC codes design for 2D ISI channels," in *Proc. Digest APMRC*, Singapore, Oct./Nov. 2012, pp. 1–2.

[16] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[17] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, 2001.

[18] T. Richardson and R. Urbanke, *Modern Coding Theory*. USA: Cambridge University Press, 2008.

[19] D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug 2006.

[20] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Braided convolutional codes with sliding window decoding," *IEEE Trans. Communications*, vol. 65, no. 9, pp. 3645–3658, 2017.

[21] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[22] C. Measson, A. Montanari, T. J. Richardson, and R. Urbanke, "The generalized area theorem and some of its consequences," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4793–4821, 2009.

[23] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Wave-like solutions of general 1-D spatially coupled systems," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4117–4157, 2015.

Paper IV

Low-Density Parity-Check Codes and Spatial Coupling for Quantitative Group Testing

A non-adaptive quantitative group testing (GT) scheme based on sparse codes-on-graphs in combination with low-complexity peeling decoding was introduced and analyzed by Karimi *et al.*. In this work, we propose a variant of this scheme based on low-density parity-check codes where the BCH codes at the constraint nodes are replaced by simple single parity-check codes. Furthermore, we apply spatial coupling to both GT schemes, perform a density evolution analysis, and compare their performance with and without coupling. Our analysis shows that both schemes improve with increasing coupling memory, and for all considered cases, it is observed that the LDPC code-based scheme substantially outperforms the original scheme. Simulation results for finite block length confirm the asymptotic density evolution thresholds.

Keywords: Quantitative Group testing, Spatial Coupling, Low density parity check codes, Generalized Low density parity check codes

1 Introduction

The general goal of group testing (GT) [1] is to identify the set of k defective items among a population of n items by efficiently pooling groups of items in order to reduce the total number of required tests $m < n$. In the sub-linear regime [2], where the prevalence $\gamma = k/n$ tends to zero as n increases, it has been demonstrated that sparse codes-on-graphs [3], can identify all defective items with high probability with low-complexity iterative (peeling) decoding [4, 5]. In [6] and [7], this idea was extended from non-quantitative to quantitative GT using t-error correcting BCH codes at the constraint nodes of a generalized low-density parity-check (GLDPC) code with regular and irregular variable node (VN) degrees, respectively. It turns out that the strongest codes, with largest VN degree d_{v} and decoding radius t, do not perform best with iterative decoding. Instead, the minimum number of required tests in [6, 7] is achieved for $t = 2$ and the distribution of d_{v} has to be chosen carefully for every t.

Spatial coupling of regular graphs is an attractive alternative to the sensitive optimization of irregular graphs, thanks to the threshold saturation phenomenon that leads to robust performance with iterative decoding even for large d_{v}. First observed for low-density parity-check (LDPC) codes [8,9], this behavior extends to other graph-based systems such as GLDPC codes [10] or iterative decoding and detection [11, 12]. To the best of our knowledge, however, the concept of spatial coupling has never been applied to GT schemes.

Our main contribution in this paper is two-fold: first, we propose a novel quantitative GT scheme based on LDPC codes as an alternative to the GLDPC code-based GT scheme in [6, 7]. A corresponding peeling decoder is presented, which cannot rely on local error correction of the component codes (since $t = 0$) but instead takes advantage of the cases where either all or none of the items within a test are defective. Second, we apply spatial coupling to both schemes. We further perform a density evolution analysis of the LDPC code-based GT scheme and of the coupled schemes to investigate the effect of increasing coupling memory for various combinations of d_v and t.

We consider two scenarios for evaluating the schemes. In the first scenario, we fix the proportion of defective items γ (prevalence) and compute the minimum required rate Ω, defined as the number of tests per item. This allows for a comparison with the results presented in [6]. In the second scenario, in order to study threshold saturation, we consider a fixed graph structure with rate Ω and analyze how much γ can be increased while still maintaining reliable recovery of the items. For both scenarios, it can be observed that spatial coupling improves the performance as the coupling memory w increases. In particular, the best thresholds γ_{th} are achieved for larger values of d_{v}. We prove theoretically that threshold saturation occurs in both scenarios. Remarkably, the density evolution analysis also shows that the proposed LDPC code-based GT scheme significantly outperforms the GLDPC code-based GT scheme of [6, 7]. This is also true for the coupled schemes. Finally, we present finite block length simulation results for the LDPC code-based and GLDPC code-based GT schemes that confirm the behavior observed in the asymptotic analysis.

2 System Model

We consider a population of n items, each of which is defective with probability γ, referred to as the *prevalence*. We represent the n items by a binary vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i = 1$ if item i is defective and $x_i = 0$ if it is not. Vector $\boldsymbol{x}$ is unknown and the goal of the GT scheme is to infer it.

The GT consists of m tests and can be represented by an $m \times n$ test matrix $\boldsymbol{A} = (a_{i,j})$, where row i corresponds to test i, column j corresponds to item j, i.e., x_j, and $a_{i,j} = 1$ if item j participates in test i and $a_{i,j} = 0$ otherwise.

Here, we consider noiseless, non-adaptive quantitative group testing, where the result of each test correctly gives the number of defective items in the test. The result of the i-th test, denoted by s_i, is therefore given by

$$s_i = \sum_{j=1}^{n} x_j a_{i,j} \, .$$

We collect the results of the m tests in the *syndrome* vector $\boldsymbol{s} = (s_1, \ldots, s_m)$. It holds

$$\boldsymbol{s} = \boldsymbol{x}\boldsymbol{A}^{\mathrm{T}} \, .$$

Based on the syndrome, the goal of GT is to estimate $\boldsymbol{x}$ via a decoding operation.

The assignment of items to tests can be conveniently represented by a bipartite graph consisting of n variable nodes (VNs) corresponding to the n items and m constraint nodes (CNs) corresponding to the m tests. An edge between VN j, v_j, and CN i, c_i, is drawn if item x_j participates in test i, i.e., if $a_{i,j} = 1$.

Fig. 1 shows the bipartite graph corresponding to a scenario with 6 items and 3 tests with assignment matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \, . \tag{1}$$

The bipartite graph representation of quantitative GT traces a connection with codes-on-graphs. Hence, the theory of codes-on-graphs can be used to design good test matrices $\boldsymbol{A}$ and analyze their properties.

3 Preliminaries: Group Testing Based on GLDPC Codes

The work [6] introduced a quantitative group testing scheme based on regular GLDPC codes where the test matrix $\boldsymbol{A}$ corresponds to the partity-check matrix of a GLDPC code. Particularly, the construction in [6] is as follows. Consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph with n VNs and m_{B} CNs and its corresponding $m_{\mathsf{B}} \times n$ adjacency matrix $\boldsymbol{B}$. To construct the test matrix $\boldsymbol{A}$, each of the d_{c} non-zero elements in a row of $\boldsymbol{B}$ is replaced by a column of an $n_{\mathsf{u}} \times d_{\mathsf{c}}$ signature matrix $\boldsymbol{U} = \left(\mathbf{1}_{1\times d_{\mathsf{c}}}^{\top}, \boldsymbol{H}_{\mathrm{T}}\right)_{\mathrm{T}}$, where $\mathbf{1}_{1\times d_{\mathsf{c}}}$ is a $1 \times d_{\mathsf{c}}$ all-ones vector and $\boldsymbol{H}$, of dimensions $t\log_2(d_{\mathsf{c}}+1) \times n$, is the parity-check matrix of a t-error correcting BCH code of length d_{c}. Hence, $n_{\mathsf{u}} = t\log_2(d_{\mathsf{c}}+1)+1$, and the total number of tests is given by $m = m_{\mathsf{B}} n_{\mathsf{u}}$. (Note that, for a GLDPC code-based GT scheme,

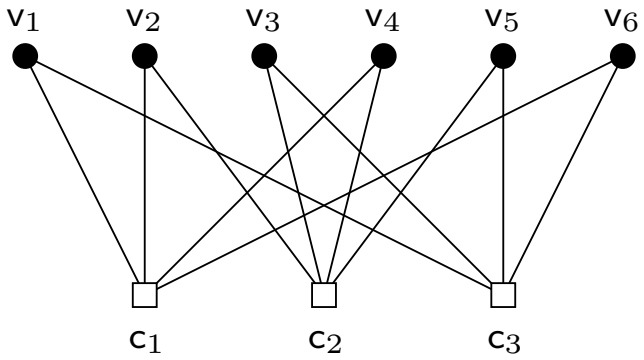


Figure 1: Bipartite graph corresponding to the assignment matrix in (1).

contrary to the bipartite graph in Fig. 1, each of the CNs corresponds to a bundle of n_{u} tests.)

We denote by Ω the *rate* of the GT scheme, i.e., the ratio between the number of tests and the number of items.[1] For the construction in [7],

$$\Omega = \frac{m}{n} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}\Big(t\left\lceil \log_2(d_{\mathsf{c}}+1)\right\rceil + 1\Big), \tag{2}$$

where the ceiling function $\lceil . \rceil$ takes care of cases where $d_{\mathsf{c}}+1$ is not a power of two.

Decoding to recover $\boldsymbol{x}$ is performed via peeling decoding, where at each iteration, due to the t-error correcting capability of the BCH codes, a CN connected to t or less unresolved defective items can identify them and their adjacent edges are peeled off the graph.

The probability that a defective item remains unidentified over iterations can be tracked via density evolution. Let $p^{(\ell)}$ the probability that a defective item remains unidentified at iteration ℓ and $q^{(\ell)}$ the probability that a CN is resolved at iteration ℓ. The quantities $p^{(\ell)}$ and $q^{(\ell)}$ are given by the following density evolution equations [7],

$$\begin{aligned} q^{(\ell)} =& \sum_{i=0}^{t-1} \binom{d_{\mathsf{c}}-1}{i} \left(p^{(\ell-1)}\right)^i \left(1-p^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \\ p^{(\ell)} =& \gamma\left(1-q^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}. \end{aligned}$$

4 Quantitative Group Testing Based on LDPC Codes

In this section, we propose a novel GT scheme based on LDPC codes in which the test matrix $\boldsymbol{A}$ is the parity-check matrix of an LDPC code or, correspondingly is obtained from the bipartite graph of an LDPC code (as the one in Fig. 1).

4.1 Proposed GT Scheme

We consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph, where each VN is connected to d_{v} CNs and each CN is connected to d_{c} VNs. The rate of the LDPC code-based

[1] Note that, interpreting $\boldsymbol{A}$ as the parity-check matrix of a code, $\Omega = 1 - R$, where R is the code rate.

GT scheme is $\Omega = \frac{d_v}{d_c}$, which can also be obtained from (2) by setting $t = 0$.

Similar to LDPC codes over the binary erasure channel and GLDPC code-based GT, decoding can be performed via peeling decoding. Peeling decoding gives rise to a sequence of residual graphs. Decoding is successful if eventually the decoder manages to peel off all VNs from the original graph, resulting in an empty graph.

Let $d_{\mathsf{c}}^{(\ell)}$ the degree of a generic CN c at iteration ℓ and $\boldsymbol{s}^{(\ell)}$ the corresponding syndrome (after the contribution of the resolved items in previous iterations has been removed). The proposed peeling decoding algorithm is based on the following observation: If $\boldsymbol{s}^{(\ell)} = 0$, then all VNs connected to c are non-defective and can be resolved. Furthermore, if $\boldsymbol{s}^{(\ell)} = d_{\mathsf{c}}^{(\ell)}$, then all VNs connected to c are defective and can also be resolved. Otherwise, none of the connected VNs can be resolved by considering c. This observation yields to the following peeling decoding algorithm:

1. Initialization: Set $s_i^{(\ell)} = s_i$ and all items as *unresolved*
2. For $\ell \geq 1$: For each CN c_i in the residual graph at iteration ℓ,
   - If $s_i^{(\ell)} = 0$ declare all connected VNs as non-defective items and peel off their adjacent edges
   - If $s_i^{(\ell)} = d_{\mathsf{c}_i}^{(\ell)}$ declare all connected VNs as defective items, subtract 1 from the syndrome of their neighboring CNs, and peel off their adjacent edges
3. If the resulting residual graph is empty or no edges have been peeled off in Step 2 (i.e., decoding stalls), stop the decoding. Otherwise, increase ℓ and return to 2)

4.2 Density Evolution

In this section, we derive the density evolution equations of the peeling decoding algorithm introduced in this section. For convenience, we group the VNs into two classes, the class of VNs corresponding to defective items, which we call *defective* VNs, and the class of VNs corresponding to non-defective items, which we call *non-defective* VNs.

Let $p_0^{(\ell)}$ be the probability that a message from a non-defective VN to a CN at iteration ℓ is an *unresolved* message, and $p_1^{(\ell)}$ the probability that a message from a defective VN to a CN at iteration ℓ is *unresolved*. Also let $q_0^{(\ell)}$ be the probability that a message from a CN to a non-defective VN is a *resolved* message, and $q_1^{(\ell)}$ be the probability that a message from a CN to a defective VN is *resolved*.

Proposition 1. *The quantities $p_0^{(\ell)}$, $p_1^{(\ell)}$, $q_0^{(\ell)}$, and $q_1^{(\ell)}$ are given by the following*

density evolution equations:

$$q_0^{(\ell)} = \sum_{i=0}^{d_{\mathsf{c}}-1} \binom{d_{\mathsf{c}}-1}{i} \gamma^i (1-\gamma)^{d_{\mathsf{c}}-1-i} \left(1 - p_1^{(\ell-1)}\right)^i \tag{3}$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_{\mathsf{c}}-1} \binom{d_{\mathsf{c}}-1}{i} \gamma^i (1-\gamma)^{d_{\mathsf{c}}-1-i} \left(1 - p_0^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \tag{4}$$

$$p_0^{(\ell)} = \left(1 - q_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \tag{5}$$

$$p_1^{(\ell)} = \left(1 - q_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} . \tag{6}$$

Proof. The probability that i out of the $d_{\mathsf{c}} - 1$ VNs connected to CN through its adjacent edges except the one on which the outgoing message is sent are defective, is given by a binomial distribution with parameters $d_{\mathsf{c}} - 1$ and γ, $\mathsf{BN}\left(() d_{\mathsf{c}} - 1, \gamma\right)$.

A message from a CN c to a non-defective VN is resolved if all incoming messages from defective VNs are resolved or all VNs connected to c are non-defective. If the number of defective items connected to c is i, then, this occurs with probability $(1 - p_1^{(\ell-1)})^i$. Considering that i is binomially distributed and summing over all i, we obtain (3).

Similarly, a message from a CN c to a defective VN is resolved if all incoming messages from non-defective VNs are resolved or all VNs connected to c are defective (i.e, $i = d_{\mathsf{c}} - 1$), yielding (4). Finally, a message from a non-defective or defective VN to a CN is unresolved if all its incoming $d_{\mathsf{c}} - 1$ messages are unresolved, yielding (5) and (6). ■

5 Group Testing with Spatial Coupling

In this section, we apply the concept of spatial coupling to the LDPC code-based GT scheme introduced in the previous section and the GLDPC code-based GT proposed in [6].

5.1 Group Testing Based on Spatially-Coupled LDPC Codes

Similar to SC-LDPC codes, the Tanner graph of a *terminated* SC-LDPC code-based GT is constructed by placing L copies of the bipartite graph of a $(d_{\mathsf{v}}, d_{\mathsf{c}})$-regular LDPC code-based GT in L spatial positions, each consisting of n_{b} VNs and m CNs. We refer to L as the coupling length and to n_{b} as the component code block length. The L copies are then coupled as follows: each VN at spatial position $\tau \in [L]$ is connected to d_{v} CNs at positions in the range $[\tau, \tau + w]$, where w is referred to as the coupling memory. For each connection, the position of the CN is uniformly and independently chosen from that range. Further, each CN at spatial position $\tau \in [L]$ is connected to d_{c} CNs at positions in the range $[\tau, \tau - w]$.

As for SC-LDPC codes, the lower degree of the CNs at the boundaries of the coupled chain yield to a wave-like decoding effect where a decoding wave propagates from the boundaries of the chain inward.

The rate of the SC-LDPC code-based GT scheme is

$$\Omega_{\mathsf{SC}} = \left(1 + \frac{w}{L}\right) \Omega \,, \tag{7}$$

with $\Omega = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}$. Note that coupling implies an increase in the number of tests by a factor of $\frac{w}{L}$ compared to the uncoupled case—akin to the rate loss of SC-LDPC codes—that vanishes as L becomes large.

The density evolution equations for SC-LDPC code-based GT are given in the following proposition.

Proposition 2. *The quantities $p_{0,\tau}^{(\ell)}$, $p_{1,\tau}^{(\ell)}$, $q_{0,\tau}^{(\ell)}$, and $q_{1,\tau}^{(\ell)}$ are given by the following density evolution equations:*

$$\begin{aligned}
q_{0,\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \sum_{i=0}^{d_{\mathsf{c}}-1} \mathsf{BN}\left(() \, d_{\mathsf{c}} - 1, i, \gamma\right) \left(1 - p_{1,\tau-j}^{(\ell-1)}\right)^{i} \\
q_{1,\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \sum_{i=0}^{d_{\mathsf{c}}-1} \mathsf{BN}\left(() \, d_{\mathsf{c}} - 1, i, \gamma\right) \left(1 - p_{0,\tau-j}^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \\
p_{0,\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \left(1 - q_{0,\tau+j}^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \\
p_{1,\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \left(1 - q_{1,\tau+j}^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \,.
\end{aligned}$$

5.2 Group Testing based on Spatially-Coupled GLDPC Codes

The coupling of GLDPC code-based GT is performed in a similar way as for SC-LDPC code-based GT. However, contrary to SC-LDPC code-based GT, which is obtained by coupling the bipartite graph corresponding to the test matrix $\boldsymbol{A}$, the coupling of GLDPC code-based GT is performed over the bipartite graph corresponding to the adjacency matrix $\boldsymbol{B}$.

The rate of the SC-GLDPC code-based GT scheme is also given by (7), with Ω given in (2).

The density evolution equations for SC-GLDPC code-based GT are given in the following proposition.

Proposition 3. *The quantities $q_{\tau}^{(\ell)}$, and $p_{\tau}^{(\ell)}$ are given by the following density evolution equations:*

$$\begin{aligned}
q_{\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \sum_{i=0}^{t-1} \binom{d_{\mathsf{c}}-1}{i} \left(p_{\tau-j}^{(\ell-1)}\right)^{i} \left(1 - p_{\tau-j}^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \\
p_{\tau}^{(\ell)} =& \frac{1}{w+1} \sum_{j=0}^{w} \gamma \left(1 - q_{\tau+j}^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \,.
\end{aligned}$$

Table 1: Ω_{th} for $\gamma = 0.15\%$ with GLDPC code-based group testing

		coupling memory				
t	d_{v}	$w = 0$	$w = 1$	$w = 2$	$w = 5$	$w = 10$
1	2	3.3588	3.3574	3.3564	3.3564	3.3564
	3	2.2374	1.9968	1.9956	1.9956	1.9956
	4	2.3715	2.0432	2.0328	2.0320	2.0320
2	2	2.2472	2.1286	2.1277	2.1268	2.1268
	3	2.4574	1.9506	1.9310	1.9310	1.9310
	4	2.8612	2.1726	2.1268	2.0650	2.0650
3	2	2.1926	1.9655	1.9639	1.9629	1.9623
	3	2.7106	2.1056	2.0443	2.0415	2.0408
	4	3.3713	2.2504	2.0637	2.0369	2.0364
5	2	2.4079	2.0580	2.0367	2.0364	2.0364
	3	3.0622	2.3407	2.1884	2.1686	2.1686
	4	3.7795	2.2653	2.2655	2.1691	2.1691

6 Numerical Results

6.1 Density Evolution Thresholds

The density evolution equations derived in Sections 4.2, 5.1, and 5.2 can be used to analyze the behavior of GT in the limit of large n, and more precisely to compute the GT *threshold*. In particular, for a fixed prevalence γ, the GT threshold Ω_{th} is defined as the minimum rate—the minimum number of tests per item—required for perfect detection of the defective items. Conversely, for a fixed rate Ω, the GT threshold γ_{th} is defined as the maximum prevalence that allows perfect detection of the defective items.[2]

Here, we give density evolution results for the proposed LDPC code-based and spatially-coupled GT schemes and compare them with the GLDPC code-based scheme in [6].

In Table 1, we give Ω_{th} for a prevalence $\gamma = 100/2^{16}$ for GLDPC code-based GT.[3] The uncoupled case, $w = 0$, corresponds to the scheme in [6]. We can see that coupling improves the threshold Ω (except for $t = 1$ and $d_{\mathsf{v}} = 2$), and the improvement increases with increasing t and d_{v}. For both the uncoupled and coupled cases, the best threshold is obtained for $t = 3$ and $d_{\mathsf{v}} = 2$.

In Fig. 2, we plot threshold Ω_{th} as a function of the prevalence γ (both in percentage) for the proposed LDPC code-based GT scheme with $d_{\mathsf{v}} = 5$, the GLDPC code-based GT scheme of [6] with $t = 2$ and $d_{\mathsf{v}} = 2$, and the coupled versions of both. We observe that the LDPC code-based GT scheme significantly outperforms the scheme in [6]. Spatial coupling improves Ω_{th} for both schemes, with the largest improvement for the proposed LDPC code-based GT scheme.

[2]We consider the threshold Ω_{th}, as this is the quantity considered in [6]. However, from a coding perspective, it is interesting to fix the rate of the scheme and compute γ_{th}, which is akin to the belief propagation threshold for codes-on-graphs.

[3]We use this prevalence as it is the one considered in [6, Fig. 2]

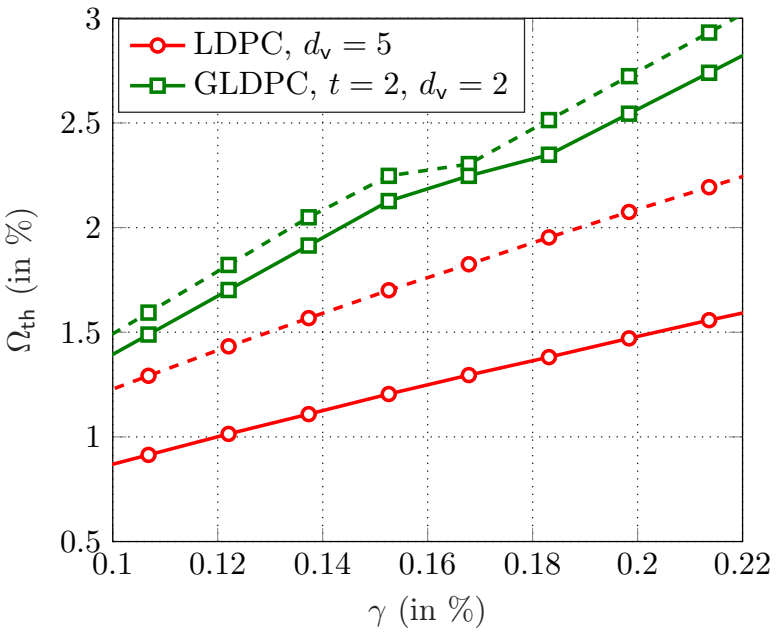


Figure 2: Ω_{th} as a function of γ for LDPC code-based and GLDPC code-based schemes. Dashed lines are for the uncoupled schemes, while solid lines are for the coupled schemes.

Finally, in Tables 2 and 3, we give γ_{th} for $\Omega = 5\%$ for GLDPC code-based and LDPC code-based GT, respectively. For LDPC code-based GT (Table 3), we observe that with coupling the threshold improves with increasing d_{v} (similarly to LDPC codes). Compared to GLDPC code-based GT (Table 2), LDPC code-based GT achieves significantly higher thresholds. Furthermore, we generally observe that the thresholds tend to converge to a constant value for a large enough coupling memory w.

6.2 Simulation Results

In this section, we give simulation results for finite block length. In Fig. 3, we plot the misdetection rate, i.e., the fraction of defective items not identified, as a function of the prevalence γ for the proposed LDPC code-based GT scheme with no coupling (dashed lines) and coupling (solid lines) for $d_{\mathsf{v}} = 3$, 5, and 10, and rate $\Omega = 0.05$. The block length of the uncoupled scheme is $n = 153000$. For the coupled scheme, we consider $w = 5, L = 200$, and component code block length $n_{\mathsf{b}} = 102000$. Further, the coupled scheme is decoded by iterating on the entire chain (not using a window decoder). We observe that coupling significantly improves performance, particularly for large d_{v}, in agreement with the density evolution results (cf. Table 3). The density evolution thresholds are given by the vertical lines. We remark that the latency (defined as the number of items that need to be tested before delivering test results) of the coupled scheme is much larger than that of the uncoupled scheme. The former is $n_{\mathsf{b}}L$, while the latter is n. Note, however, that increasing n for the uncoupled scheme marginally improves its performance (the limit is given by the density evolution threshold), hence the figure highlights how much one can gain with coupling if

Table 2: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based Group Testing

		coupling memory				
t	d_{v}	$w = 0$	$w = 1$	$w = 2$	$w = 5$	$w = 10$
1	2	0.2487	0.2502	0.2502	0.2502	0.2502
	3	0.3708	0.4166	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425	0.4425
2	2	0.3983	0.4257	0.4257	0.4257	0.4257
	3	0.3372	0.4242	0.4288	0.4288	0.4288
	4	0.2884	0.4120	0.4318	0.4333	0.4333
3	2	0.3784	0.4211	0.4227	0.4227	0.4227
	3	0.3189	0.4257	0.4379	0.4379	0.4395
	4	0.2441	0.3662	0.3983	0.4028	0.4028
5	2	0.3418	0.3998	0.4044	0.4044	0.4044
	3	0.2686	0.3784	0.4044	0.4089	0.4089
	4	0.2014	0.3159	0.3616	0.3769	0.3769

Table 3: γ_{th} for $\Omega = 5\%$ with LDPC Code-Based Group Testing

	coupling memory				
d_{v}	$w = 0$	$w = 1$	$w = 2$	$w = 5$	$w = 10$
3	0.4555	0.5544	0.5508	0.5559	0.5559
4	0.5982	0.8423	0.8532	0.8540	0.8540
5	0.6416	0.9682	1.0270	1.0274	1.025
6	0.6464	1.0044	1.1196	1.1325	1.1327
7	0.6353	0.9999	1.1585	1.1978	1.1980
10	0.5773	0.9188	1.1272	1.2814	1.2816

latency is not a problem.

In Fig. 4, we plot the misdetection rate for uncoupled (dashed lines) and coupled (solid lines) GLDPC code-based GT with $t = 3$, $d_{\mathsf{v}} = 3$, and $\Omega = 0.05$. Further, we consider two component code block lengths, $n_{\mathsf{b}} = 153000$ and $n_{\mathsf{b}} = 10200$. For $n_{\mathsf{b}} = 153000$, we assume full decoding, while for $n_{\mathsf{b}} = 10200$ we assume a sliding window (SW) decoding [13] with window size $W = 15$. The latency of the coupled scheme is therefore $n_{\mathsf{b}}W$, i.e., the latencies of the uncoupled scheme and coupled scheme with SW decoding are identical. Notably, for SW decoding and the same latency, coupling still outperforms the uncoupled scheme. As for the LDPC code-based GT scheme, coupling with full decoding significantly improves performance.

In Fig. 5, we consider the scenario with fixed prevalence γ and varying rate Ω, corresponding to [6, Fig. 2]. For GLDPC code-based GT with $t = 2$ and $d_v = 2$, we plot the misdetection rate as a function of the number of tests per defective item. Each point in this plot corresponds to a different rate Ω. We observe for the uncoupled scheme (dashed) that the curve flattens as the number

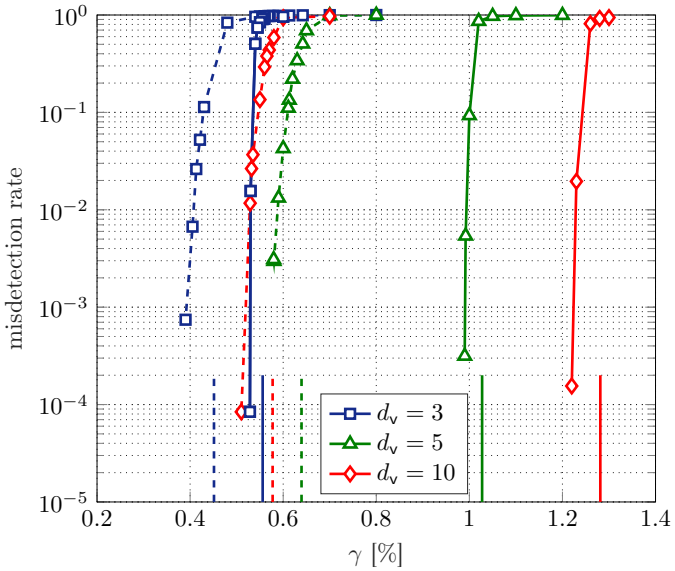


Figure 3: Misdetection rate for uncoupled (dashed) and coupled (solid) LDPC code-based GT.

of tests increases, while it decays very steeply for the coupled scheme (solid).

7 Conclusions and Discussion

Our results show that the scheme we proposed, which uses LDPC , outperforms the the scheme using GLDPC codes. This means for the same fraction of defectives items the LDPC scheme requires less tests than the GLDPC scheme. The same is true also if we consider a fixed number of tests and estimate how many defective items can we resolve. The good performance of the LDPC codes comes from the fact that the it does not use any local error correction capability i.e $t = 0$, thus each test can be processed alone. This loss of local error correction is more than compensated by the performance of the overall system when all tests interact. This is not true for the GLDPC scheme which uses local error correction capability meaning that the decoder needs a bin of $t \log_2(d_c + 1) + 1$ tests to resolve at most t defective items. This penalty in the number of tests per test bin is not compensated by the performance of the overall system.

We have also demonstrated that threshold saturation occurs when spatial coupling is applied to group testing. We prove this by showing that the recursion for group testing with LDPC scheme constitutes a vector admissible system. The computed potential threshold together with finite length simulation results show that spatial coupling improves the performance of group testing. We get not only improvement in thresholds but also the error floor is reduced when blocks of finite length are used.

We also observe that spatial coupling provides consistency in performance with increasing variable node degree, d_v. Without coupling, performance improves as d_v is increased from 3 to 6 but starts to degrade for high values. With coupling, however, the performance is consistently improving with d_v. This is because the performance without coupling is determined by the BP decoder

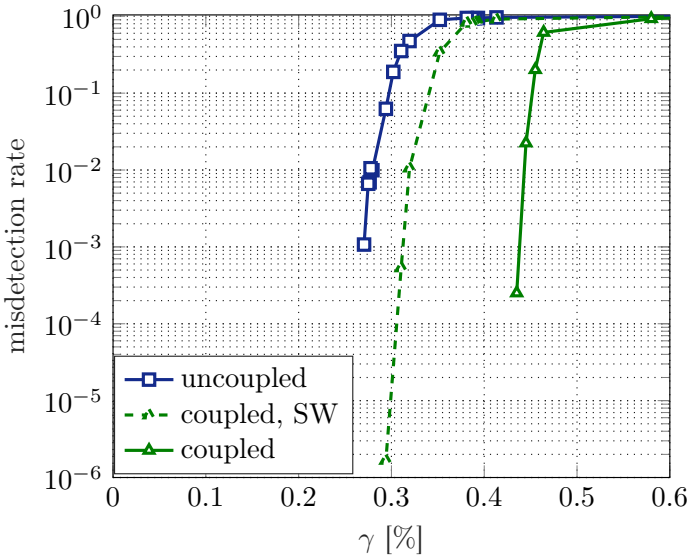


Figure 4: Misdetection rate for uncoupled (dashed) and coupled (solid) GLDPC code-based GT with $t = 3$ and $d_{\mathrm{v}} = 3$.

which is has its limitations when d_{v} is increased. The MAP threshold, on the other hand, improves with d_{v} and coupling approaches the performance of a MAP decoder hence the consistency.

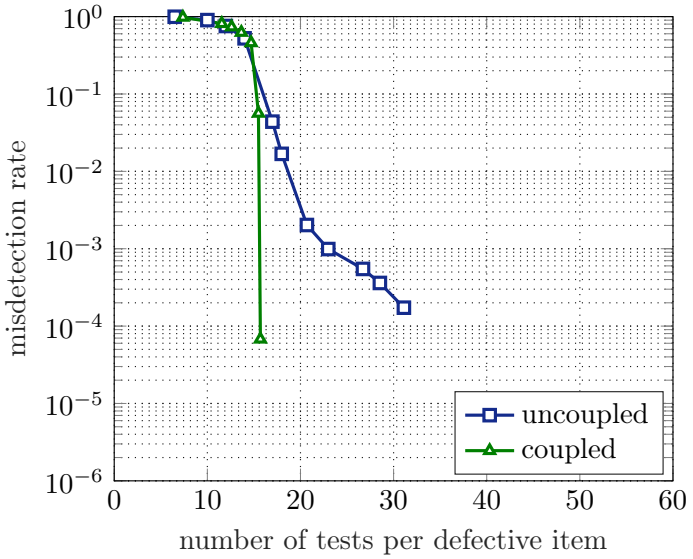


Figure 5: Misdetection rate as a function of the number of tests per defective item for GLDPC code-based GT with $t = 2$ and $d_{\mathrm{v}} = 2$, and $\gamma = 0.15\%$.

References

[1] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[2] O. Gebhard, M. Hahn-Klimroth, D. Kaaser, and P. Loick, "Quantitative group testing in the sublinear regime," *CoRR*, vol. abs/1905.01458, 2019. [Online]. Available: http://arxiv.org/abs/1905.01458

[3] T. Wadayama, "An analysis on non-adaptive group testing based on sparse pooling graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 2013.

[4] K. Lee, R. Pedarsani, and K. Ramchandran, "SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July 2016.

[5] A. Vem, N. T. Janakiraman, and K. R. Narayanan, "Group testing using left-and-right-regular sparse-graph codes," *CoRR*, vol. abs/1701.07477, 2017. [Online]. Available: http://arxiv.org/abs/1701.07477

[6] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *2019 IEEE Inf. Theory Workshop (ITW)*, 2019.

[7] ——, "Non-adaptive quantitative group testing using irregular sparse graph codes," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019.

[8] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

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

[10] D. G. M. Mitchell, P. M. Olmos, M. Lentmaier, and D. J. Costello, "Spatially coupled generalized LDPC codes: Asymptotic analysis and finite length scaling," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 3708–3723, 2021.

[11] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Ottawa, Canada, June 2012.

[12] M. M. Mashauri, A. Graell i Amat, and M. Lentmaier, "On the universality of spatially coupled LDPC codes over intersymbol interference channels," in *Proc. IEEE Information Theory Workshop (ITW)*, Kanazawa, Japan, Oct. 2021.

[13] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.

Paper V

Spatially Coupled LDPC and GLDPC Codes for Quantitative Group Testing

We recently proposed a quantitative group testing (GT) scheme with low-complexity peeling decoding based on low-density parity-check (LDPC) codes. Based on finite length simulations and a density evolution analysis we were able to demonstrate that simple $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC codes can be more efficient for GT than existing generalized LDPC (GLDPC) code constructions based on BCH component codes. Even larger gains were numerically observed in combination with spatial coupling. In this paper, we use vector admissible systems to prove threshold saturation and compute the corresponding potential thresholds. We furthermore prove threshold saturation for the GLDPC scheme and obtain the corresponding potential thresholds which still show a gap to the LDPC scheme. We observe that one reason for this comes from the sub-optimal decoder used in the GLDPC scheme and suggest ways to improve it by exploiting the LDPC decoder.

Keywords: Quantitative Group testing, Spatial Coupling, Low density parity check codes, Threshold Saturation, Vector admissible system

---

Mgeni Makambi Mashauri, Alexandre Graell i Amat, and Michael Lentmaier, 'LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet", *To be submitted to IEEE Transactions on Information Theory.*

1 Introduction

Group testing (GT) is a technique of efficiently identifying items of interest (which we call defective items) in a population by testing items in groups. With GT much fewer tests are needed to successfully identify all defective items compared to naive individual testing of items, especially if the number of defective items is much lower than the population size n. GT can be classified based on the test results as quantitative or non-quantitative. With quantitative GT each test results indicates how many of the items in the test are defective while for non-quantitative GT the test results indicated if at least one item is defective. Over variants exists between the two extremes such as threshold GT [1] where the test result is positive if the number of defective items is greater than a predefined threshold. Another variation which is semi-quantitative GT where the test result indicates a range of possible number of defective items [2]. The problem of GT has a close connection to the problem of error correcting codes. This has led to the application of various tools from coding theory to GT, one being the use of sparse codes-on-graphs. It has been demonstrated that sparse codes-on-graphs, in combination with low complexity peeling decoding, are able to identify all defective items with high probability for both non-quantitative [3,4] and quantitative GT [5,6]. We consider noiseless non-adaptive, quantitative GT, in which the result of each test shows the exact number of defective items.

In a previous work [7], we proposed a novel peeling decoder for GT that allowed us to use simple low-density parity-check (LDPC) codes instead of generalized LDPC (GLDPC) codes based on t-error correcting codes [5, 6].

Despite of losing the local error correction capability in this construction, it is possible to take advantage of two extreme scenarios: one when all items connected to a test are non-defective, and the other when all items connected to a test are defective. Based on this we were able to show that LDPC codes, with $t = 0$, are more efficient for GT than GLDPC codes with $t > 0$. As shown in Fig. 1, the LDPC scheme requires much fewer tests than the GLDPC scheme for the same number of defective items k. For example, for $k = 800$ the GLDPC scheme requires slightly more than 8700 tests while the LDPC scheme requires around 5400 tests. The gap is widening with increasing k. Furthermore, we were able to show in [7] that the performance can be improved further by applying spatial coupling. The improvement increased with coupling memory w, reaching a relatively stable value for higher w. These numerical results suggested that threshold saturation may occur for GT with LDPC codes.

In this work, we prove that threshold saturation indeed occurs for the quantitative GT scheme based on LDPC codes. The proof is done by showing that the density evolution (DE) recursions for GT with LDPC codes satisfy the conditions for being a vector admissible system [8]. We also show that threshold saturation occurs for quantitative GT with GLDPC codes. Furthermore we highlight some of the reasons why GLDPC based GT does not work well the main one being the suboptimality of the bounded minimum distance (BMD) decoder. We provide ways to improve the decoder for the GLDPC.

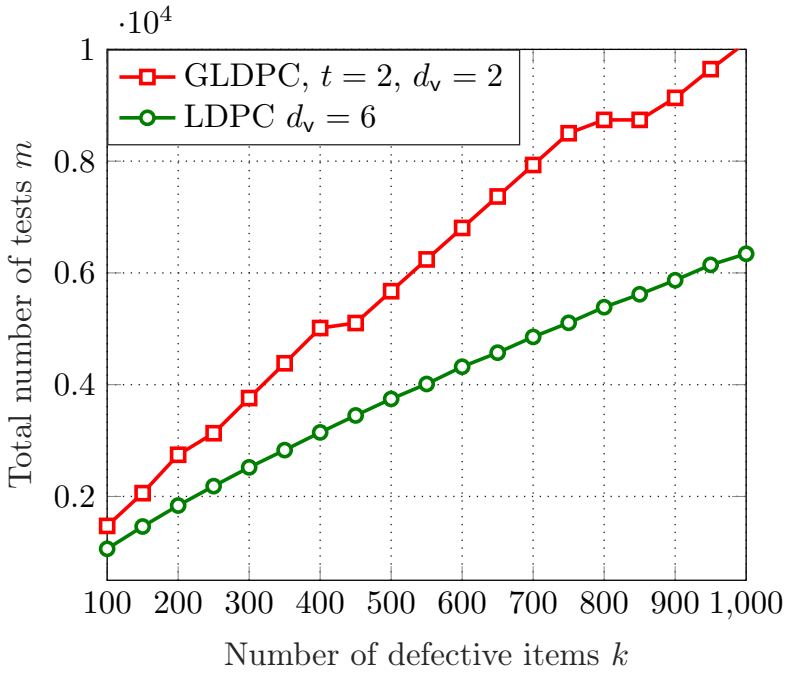


Figure 1: A comparison of GT based on LDPC- and GLDPC codes, showing the total number of tests m required for a population size $n = 2^{16}$.

2 Preliminaries

2.1 System Model

We consider a population of n items represented by a binary vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i = 1$ if item i is defective and $x_i = 0$ if it is not defective. Each item is defective with probability γ. A GT scheme aims at recovering $\boldsymbol{x}$ using m tests where $m < n$.

A GT scheme can be represented by an $m \times n$ adjacency matrix $\boldsymbol{A} = (a_{i,j})$, where $a_{i,j} = 1$ if item j participates in test i and $a_{i,j} = 0$ otherwise. We consider quantitative GT without noise, where the result of each test gives the exact number of defective items participating in the test. Collecting the results of all tests in a vector $\boldsymbol{s} = (s_1, \ldots, s_m)$, called the syndrome, we have

$$s(\mathsf{c}) = \boldsymbol{x} A^{\mathsf{T}} .$$

The assignment of items to tests can be conveniently represented by a bipartite graph consisting of n variable nodes (VNs) corresponding to the n items and m constraint nodes (CNs) corresponding to the m tests. An edge between VN j, and CN i is drawn if item x_j participates in test c_i, i.e., if $a_{i,j} = 1$.

Fig. 2 shows the $(2, 4)$ bipartite graph corresponding to a scenario with $n = 6$ items and $m = 3$ tests with assignment matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} . \tag{1}$$

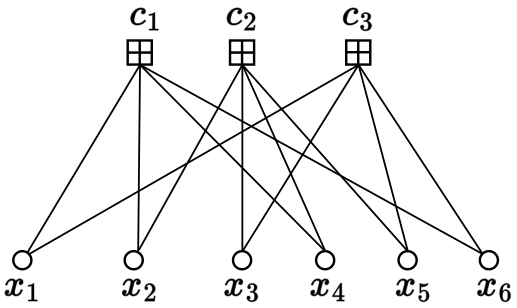


Figure 2: Bipartite graph corresponding to the assignment matrix in (1).

We define the ratio of the number of tests m to the population size n as the rate Ω. That is

$$\Omega = \frac{m}{n}\,.$$

2.2 Two Different Performance Measures

We consider two scenarios for evaluating the performance of a GT scheme. In the first scenario, we fix the proportion γ of defective items and evaluate the minimum rate Ω_{th} at which all defective items can be detected with high probability. This corresponds to the conventional GT perspective, where the number of defective items is fixed and the aim is to reduce the total number of tests required for successful decoding. In the second scenario, we fix the rate Ω and determine the highest fraction γ_{th} of defectives that can be tolerated for successful decoding. This scenario was first introduced in [7] and is more closely related to the threshold definition in channel coding, where the graph is fixed in terms of node degrees. In some applications, like multi-access communication [9, 10], this corresponds to asking the question how much traffic a network can tolerate for given channel resources.

2.3 Quantitative Group Testing Based on GLDPC Codes

The work [5] introduced a quantitative group testing scheme based on regular GLDPC codes where the test matrix $\boldsymbol{A}$ corresponds to the parity-check matrix of a GLDPC code with a t-error correcting BCH code as the component code. We first briefly discuss BCH codes and then present how the GLDPC is constructed and the corresponding density evolution.

2.4 BCH Codes

Bose–Chaudhuri–Hocquenghem codes (BCH codes), named after its inventors, form a class of cyclic error-correcting codes that are constructed using polynomials over a Galois field (GF). A Galois field is a field with finite number of elements. A Galois field with q elements is denoted as GF(q). The order of an element a in a Galois field is the smallest integer r such that $a^r = 1$. An element whose order equals $q - 1$ is called a primitive element and is denoted as α. Details about Galois fields can be found in [11, 12].

The decoding of BCH codes is very efficient since it uses algebraic methods. An interesting feature of BCH codes is that one can design precisely a code

which can correct t errors. That is for every integers $m \geq 3$ and $t < 2^{m-1}$, there exists a BCH code with the following parameters,

$$\begin{aligned} &\text{Block length:} && N = 2^m - 1 \\ &\text{Number of parity check digits:} && N - K \leq mt \\ &\text{Minimum distance:} && d_{\min} \geq 2t + 1\,. \end{aligned}$$

A binary $N-$tuple $(v_0, v_1, \cdots, v_{N-1})$ is a code word of a t correcting BCH code if and only if the polynomial $\boldsymbol{v}(X) = v_0 + v_1 \cdot X + v_2 \cdot X^2 + \cdots + v_{N-1} \cdot X^{N-1}$ has $\alpha, \alpha^2, \ldots \alpha^{2t}$ as roots. This means for every codeword we have

$$\boldsymbol{v}(\alpha^i) = v_0 + v_1 \cdot \alpha^i + v_2 \cdot \alpha^{2i} + \cdots + v_{N-1}\alpha^{(N-1)i} = 0$$

for $1 \leq i \leq 2t$. In matrix notation this can be written as

$$(v_0, v_1, v_2 \cdots v_{N-1}) \cdot \begin{pmatrix} 1 \\ \alpha^i \\ \alpha^{2i} \\ \vdots \\ \alpha^{(N-1)i} \end{pmatrix} * = 0\,.$$

With this it can be seen that the parity check matrix $\boldsymbol{H}$ is given as

$$\boldsymbol{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{N-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^2)^3 & \cdots & (\alpha^2)^{N-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{N-1} \\ \vdots & & & & & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & (\alpha^{2t})^3 & \cdots & (\alpha^{2t})^{N-1} \end{pmatrix}.$$

For an element of GF(2^m) represented as $a = \alpha^i$, every element represented as $(\alpha^i)^{2^l}$ is a conjugate of a. If α^j is a conjugate of α^i for some i and j then α^j is a root of a polynomial $\boldsymbol{v}(X)$ with coefficients in GF(2) if and only if α^i is a root of $\boldsymbol{v}(X)$. We can thus omit the rows of $\boldsymbol{H}$ which correspond to conjugates of any other row. This implies that the matrix $\boldsymbol{H}$ can be reduced to the form

$$\boldsymbol{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{N-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{N-1} \\ \vdots & & & & & \vdots \\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & (\alpha^{2t-1})^3 & \cdots & (\alpha^{2t-1})^{N-1} \end{pmatrix}. \tag{2}$$

To get the binary representation of $\boldsymbol{H}$ we represent each element of GF(2^m) by its binary representation in column form.

Example 2.1. *Consider as 2-error correcting BCH code with $m = 3$. We have $N = 2^3 - 1 = 7$. The parity check matrix is given as*

$$\begin{aligned} \boldsymbol{H} &= \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \end{pmatrix} \\ &= \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \end{pmatrix}. \end{aligned}$$

Here we have used the fact that the order of the primitive element is 7*, that is* , $\alpha^7 = 1$*. The corresponding binary representation is given as*

$$\boldsymbol{H} = \left(\begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right).$$

2.5 Construction of GLDPC Based Test Matrix

The construction of a GLDPC based GT matrix is as follows [5]. Consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph with n VNs and m_{B} CNs and its corresponding $m_{\mathsf{B}} \times n$ adjacency matrix $\boldsymbol{B}$. To construct the test matrix $\boldsymbol{A}$, each of the d_{c} non-zero elements in a row of $\boldsymbol{B}$ is replaced by a column of an $n_{\mathsf{u}} \times d_{\mathsf{c}}$ signature matrix $\boldsymbol{U} = \left(\mathbf{1}_{1\times d_{\mathsf{c}}}^{\top}, \boldsymbol{H}^{\mathsf{T}}\right)^{\mathsf{T}}$, where $\mathbf{1}_{1\times d_{\mathsf{c}}}$ is a $1 \times d_{\mathsf{c}}$ all-ones vector and $\boldsymbol{H}$, of dimensions $t\log_2(d_{\mathsf{c}}+1) \times d_{\mathsf{c}}$, is the parity-check matrix of a t-error correcting BCH code of length d_{c}. Hence, $n_{\mathsf{u}} = t\log_2(d_{\mathsf{c}}+1)+1$, and the total number of tests is given by $m = m_{\mathsf{B}} n_{\mathsf{u}}$. (Note that, for a GLDPC code-based GT scheme, contrary to the bipartite graph in Fig. 2, each of the CNs corresponds to a bundle of n_{u} tests.)

Example 2.2. *Consider a regular adjacency matrix given as*

$$\boldsymbol{B} = \left(\begin{array}{cccccccccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right),$$

with $d_{\mathsf{v}} = 2$ *and* $d_{\mathsf{c}} = 7$ *for a population of* $n = 14$*. The number of CNs is then* 4*. The signature matrix for* $t = 1$ *is given as*

$$\boldsymbol{U} = \left(\begin{array}{ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right).$$

The test matrix is then given as

$$\boldsymbol{A} = \left(\begin{array}{cccccccccccccc}
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}\right)$$

We denote by Ω the *rate* of the GT scheme, i.e., the ratio between the number of tests and the number of items.[1] For the construction in [6],

$$\Omega = \frac{m}{n} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}\Big(t\left\lceil \log_2(d_{\mathsf{c}}+1)\right\rceil + 1\Big), \tag{3}$$

where the ceiling function $\lceil . \rceil$ takes care of cases where $d_{\mathsf{c}}+1$ is not a power of two.

2.6 Decoding of GLDPC Based GT

To recover $\boldsymbol{x}$, decoding is performed by each component code. Only components with t or less defective items can be decoded. This can be read form the first row of each component (corresponding to the all one row in $\boldsymbol{U}$). Decoding is done as follows. The syndrome is initiated as $\boldsymbol{s}^{(0)} = \boldsymbol{s}$.

1. Identify component codes with t or less defective items. If no component satisfying this is found the decoder halts.
2. For each component code with t or less defective items do the following
   (a) Calculate the error location polynomial $\Lambda(x)$ using the Berlekamp–Massey algorithm.
   (b) Factorize $\Lambda(x)$ to determine the error locations.
   (c) Label items indexed by the error locations as *defective*
3. Peel off resolved items form the graph
4. Recompute the syndrome $\boldsymbol{s}^{(\ell)}$ and restart from Step 1.

[1] Note that, interpreting $\boldsymbol{A}$ as the parity-check matrix of a code, $\Omega = 1 - R$, where R is the code rate.

Note that the Berlekamp–Massey algorithm [12, pg. 209] requires $2t$ syndrome alphabets in GF(2^m) but $\boldsymbol{H}$ in (2) has only t equations. The other t components can be generated by using the fact that for a polynomial $f(x)$ with coefficients in GF(2) the following holds [12, pg. 42]

$$f(\alpha^2) = (f(\alpha))^2 \ . \tag{4}$$

2.7 DE for Uncoupled GLDPC Code Based GT

The probability that a defective item remains unidentified over iterations can be tracked via density evolution. Let $p^{(\ell)}$ the probability that a defective item remains unidentified at iteration ℓ and $q^{(\ell)}$ the probability that a CN is resolved at iteration ℓ. The quantities $p^{(\ell)}$ and $q^{(\ell)}$ are given by the following density evolution equations [6],

$$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_{\mathsf{c}}-1}{i} \left(p^{(\ell-1)}\right)^i \left(1-p^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \tag{5}$$

$$p^{(\ell)} = \gamma \left(1-q^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} . \tag{6}$$

This can readily be seen from the fact that a test sends a message *resolved* to a defective VN if among the other $d_{\mathsf{c}}-1$ VNs less than t are defective and unresolved.

3 Quantitative Group Testing Based on LDPC Codes

In this section, we propose a novel GT scheme based on LDPC codes in which the test matrix $\boldsymbol{A}$ is the parity-check matrix of an LDPC code or, correspondingly is obtained from the bipartite graph of an LDPC code (as the one in Fig. 2).

3.1 Proposed GT Scheme

We consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph, where each VN is connected to d_{v} CNs and each CN is connected to d_{c} VNs. The rate of the LDPC code-based GT scheme is $\Omega = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}$, which can also be obtained from (3) by setting $t = 0$.

Similar to LDPC codes over the binary erasure channel and GLDPC code-based GT, decoding can be performed via peeling decoding. Peeling decoding gives rise to a sequence of residual graphs. Decoding is successful if eventually the decoder manages to peel off all VNs from the original graph, resulting in an empty graph.

3.2 Peeling Decoding

Let $d_{\mathsf{c}}^{(\ell)}$ be the degree of a CN c at iteration ℓ and $\boldsymbol{s}^{(\ell)}$ the corresponding syndrome. The decoding algorithm is based on the following observation: if $\boldsymbol{s}^{(\ell)} = 0$, then all VNs connected to c are non-defective and can be resolved. Furthermore, if $\boldsymbol{s}^{(\ell)} = d_{\mathsf{c}}^{(\ell)}$, then all VNs connected to c are defective and can also be resolved. Otherwise, none of the connected VNs can be resolved by

considering c. The degree $d_{\mathsf{c}}^{(\ell)}$ is updated by removing all resolved items in previous iterations while $\boldsymbol{s}^{(\ell)}$ is updated by subtracting the contribution of resolved defective items in each CN. This is summarized in Algorithm 1.

Algorithm 1 Decoding of LDPC code-based GT

INPUT: syndrome $\boldsymbol{s}$, graph G (with m tests and n items)
Output: $\boldsymbol{x}$

Initialization $\ell = 1$, $\mathsf{G}^1 = \mathsf{G}$, continue=TRUE,
$x_j = \textit{unresolved}$ for $j = 1 : n$, $s_i^{(1)} = s_i$ and $d_{\mathsf{c}_i}^{(1)} = d_{\mathsf{c}} \, \forall i$
while continue==TRUE **do**
  found=FALSE
  for $i = 1$ to m **do**
    if $s_i^{(\ell)} = 0$ **then**
      Set all items connected to c_i to 0
      Peel the items set to 0 from the graph G^{ℓ}
      found=TRUE
    else if $s_i^{(\ell)} = d_{\mathsf{c}_i}^{(\ell)}$ **then**
      Set all items connected to c_i to 1
      Peel the items set to 1 from the graph G^{ℓ}
      found=TRUE
    end if
  end for
  $\ell = \ell + 1$
  if found==FALSE or G^{ℓ} is empty **then**
    continue=FALSE
  end if
end while

3.3 DE for Uncoupled LDPC Code Based GT

It can be observed that a test or an item sends two possible message types, *resolved* or *unresolved*. If the message is *resolved* it represents the actual value of the item, i.e., 0 or 1. It was shown in [7] that the DE equations for the decoder discussed is given by

$$q_0^{(\ell)} = \sum_{i=0}^{d_{\mathsf{c}}-1} \binom{d_{\mathsf{c}}-1}{i} \gamma^i (1-\gamma)^{d_{\mathsf{c}}-1-i} \left(1 - p_1^{(\ell-1)}\right)^i \tag{7}$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_{\mathsf{c}}-1} \binom{d_{\mathsf{c}}-1}{i} \gamma^i (1-\gamma)^{d_{\mathsf{c}}-1-i} \left(1 - p_0^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i} \tag{8}$$

$$p_0^{(\ell)} = \left(1 - q_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \tag{9}$$

$$p_1^{(\ell)} = \left(1 - q_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} . \tag{10}$$

Here $q_0^{(\ell)}$ and $q_1^{(\ell)}$ are the probabilities that a CN sends a message *resolved* to a non-defective and defective VN, respectively, during iteration ℓ. While $p_0^{(\ell)}$

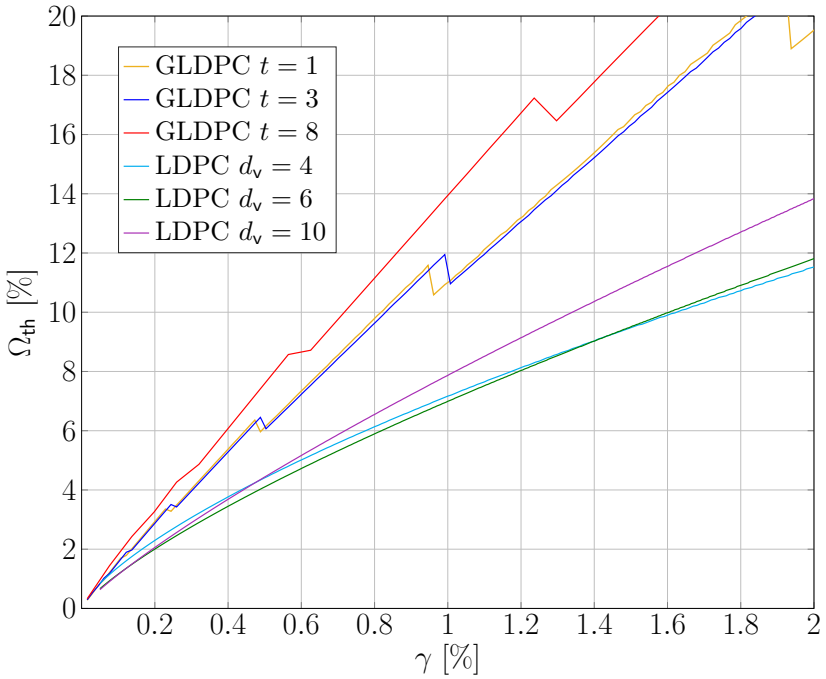


Figure 3: The minimum rate Ω_{th} for a range of γ for both GLDPC and LDPC based schemes without spatial coupling. For the GLDPC $d_{\mathsf{v}} = 2$ except for $t = 1$ where $d_{\mathsf{v}} = 3$.

and $p_1^{(\ell)}$ are the probabilities that a VN sends a message *unresolved* to a CN given that the VN is non-defective and defective, respectively. In this section we derive an alternative but equivalent set of DE equations to those in [7]. The alternative equations are easier to handle, especially for the proof of threshold saturation. Let $x_0^{(\ell)}$ be the probability that a message from a CN to a non-defective VN is *unresolved* , and $x_1^{(\ell)}$ be the probability that a message from a CN to a defective VN is *unresolved*. Also, let $y_0^{(\ell)}$ be the probability that an item is non-defective and sends a message *unresolved* to a CN, and $y_1^{(\ell)}$ be the probability that an item is defective and sends a message *unresolved* to a CN. It can be observed that $x_0^{(\ell)} = 1 - q_0^{(\ell)}$ and $x_1^{(\ell)} = 1 - q_1^{(\ell)}$, while $y_0^{(\ell)} = (1-\gamma)q_0^{(\ell)}$ and $y_1^{(\ell)} = \gamma p_1^{(\ell)}$.

Proposition 4. *The quantities $y_0^{(\ell)}$, $y_1^{(\ell)}$, $x_0^{(\ell)}$, and $x_1^{(\ell)}$ are given by the following DE equations:*

$$x_0^{(\ell)} = 1 - \left(1 - y_1^{(\ell-1)}\right)^{d_{\mathsf{c}}-1} \tag{11}$$

$$x_1^{(\ell)} = 1 - \left(1 - y_0^{(\ell-1)}\right)^{d_{\mathsf{c}}-1} \tag{12}$$

$$y_0^{(\ell)} = (1-\gamma)\left(x_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} \tag{13}$$

$$y_1^{(\ell)} = \gamma\left(x_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}. \tag{14}$$

Proof. A message from a CN c to a non-defective VN is *resolved* if none of the defective items (if any) among all $d_{\mathsf{c}} - 1$ other items sends a message *unresolved*, i.e., it is connected to zero unresolved defective items. This happens with probability $(1 - y_1^{(\ell-1)})^{d_{\mathsf{c}}-1}$. We can then compute $x_0^{(\ell)}$ as the complement of this. Similarly, a message from a CN c to a defective VN is *resolved* if it has no unresolved non-defective item among the $d_{\mathsf{c}} - 1$ other items. This happens with probability $(1 - y_0^{(\ell-1)})^{d_{\mathsf{c}}-1}$.

A non-defective item sends a message *unresolved* to a CN if all of the incoming messages from the other $d_{\mathsf{v}} - 1$ CNs are *unresolved*. Thus the probability that an item sends a message *unresolved* given that it is non-defective is given by $\left(x_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}$, which when multiplied by $(1-\gamma)$ gives $y_0^{(\ell)}$. The same reasoning applies for $y_1^{(\ell)}$. ■

Fig. 3 shows the thresholds of uncoupled LDPC codes for a range of γs without spatial coupling. The GLDPC scheme is also shown for comparison. It can be observed that the LDPC scheme outperforms the GLDPC scheme for most of the range. The performance for the LDPC scheme, however, is not consistent with increasing d_{v} for the whole range. For example, for $\gamma = 0.4\%$ we have the $d_{\mathsf{v}} = 6$ as the best followed by $d_{\mathsf{v}} = 10$ with $d_{\mathsf{v}} = 4$ being the worst. For $\gamma = 1.8$, however, $d_{\mathsf{v}} = 4$ is the best and $d_{\mathsf{v}} = 10$ the worst.

4 Quantitative GT with Spatially Coupled LDPC codes

With spatial coupling, blocks of VNs are interconnected in contrast to classical GT where each block is treated separately. This is inspired by works on spatially coupled LDPC (SC-LDPC) codes [13–15], which have shown to perform very well. Each block can be seen as occupying a spatial position τ, i.e., we have n_{b} VNs and m_{b} CNs at each spatial position. The coupling is done as follows: each VN at spatial position τ is connected to d_{v} CNs at positions in the range $[\tau, \tau + w]$ with the positions chosen uniformly at random. The parameter w is referred to as the coupling memory. Further, each CN at spatial position τ is connected to d_{c} VNs at positions in the range $[\tau, \tau - w]$. The chain is terminated after L positions, L denoting the coupling length. The degree of all VNs is kept constant while the CNs at the edges have lower degrees compared to the inner CNs. This also means that we have w more tests at the end of the chain resulting in a slight increase in the rate of the coupled scheme Ω_{SC} given as

$$\Omega_{\mathsf{SC}} = \left(1 + \frac{w}{L}\right) \Omega \,, \tag{15}$$

with $\Omega = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}$. The rate increase vanishes as L is increased. The lower degree of the CNs at the boundaries of the coupled chain yield a wave-like decoding effect where a decoding wave propagates from the boundaries of the chain inward.

4.1 DE for Spatially Coupled LDPC Code Based GT

Based on the description above, the DE equations for GT based on spatially coupled LDPC codes are given as

$$x_{0,\tau}^{(\ell)} = 1 - \frac{1}{w+1}\sum_{j=0}^{w}\left(1 - y_{1,\tau-j}^{(\ell-1)}\right)^{d_{\mathrm{c}}-1} \tag{16}$$

$$x_{1,\tau}^{(\ell)} = 1 - \frac{1}{w+1}\sum_{j=0}^{w}\left(1 - y_{0,\tau-j}^{(\ell-1)}\right)^{d_{\mathrm{c}}-1} \tag{17}$$

$$y_{0,\tau}^{(\ell)} = (1-\gamma)\frac{1}{w+1}\sum_{j=0}^{w}\left(x_{0,\tau+j}^{(\ell-1)}\right)^{d_{\mathrm{v}}-1} \tag{18}$$

$$y_{1,\tau}^{(\ell)} = \gamma\frac{1}{w+1}\sum_{j=0}^{w}\left(x_{1,\tau+j}^{(\ell-1)}\right)^{d_{\mathrm{v}}-1} . \tag{19}$$

4.2 Proof of Threshold Saturation for LDPC Based GT

In this section we provide a proof of threshold saturation for the LDPC code-based GT scheme. We first present two approaches which seem intuitive and explain why they fail and then present a third which is successful. The first two approaches use the perspective of fixed Ω which, as mentioned in Section 2.2, is more in line with threshold analysis for channel codes. We thus first convert the recursion to a scalar by a series of substitutions and try to prove that the the resulting system fulfills the conditions of being a *scalar admissible system* as highlighted in [16]. In the second approach we use a vector recursion and try to show whether the system is a *vector admissible system* [8]. The third approach uses the vector recursion with the perspective that the proportion of defectives is fixed and we change the rate Ω. We first define a scalar admissible system and a vector admissible system and proceed to present the approaches used.

Definition 1. *[16] A scalar recursion (f, g) defined by $x^{(0)} = 1$ and*

$$x^{(\ell)} = f\left(g\big(x^{(\ell-1)}\big); \varepsilon\right) .$$

parameterized by $\varepsilon \in [0,1]$, is a scalar admissible system *if $f : [0,1]\times[0,1] \to [0,1]$ and $g : [0,1] \to [0,1]$ are strictly increasing in all arguments. Also we have $f(x;0) = f(0;\varepsilon) = g(0) = 0$ and $f(x)$ and $g(x)$ have continuous second order derivatives on $[0,1]$ with $g'(x) > 0$ for $x \in (0,1)$.*

Definition 2. *[8] A vector recursion $(\boldsymbol{f}, \boldsymbol{g})$ defined by $\boldsymbol{x}^{(0)} = \mathbf{1}$ and*

$$\boldsymbol{x}^{(\ell)} = \boldsymbol{f}\Big(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)}); \varepsilon\Big) .$$

parameterized by $\varepsilon \in [0,1]$, is a vector admissible system *if the vector-valued functions $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \cdots, f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), \cdots, g_d(\boldsymbol{x})]$ are twice continuously differentiable, strictly increasing in all arguments (w.r.t. the partial order). It is also assumed that $\boldsymbol{f}(\boldsymbol{x};0) = \boldsymbol{f}(\mathbf{0};\varepsilon) = \boldsymbol{g}(\mathbf{0}) = \mathbf{0}$, that $\boldsymbol{f}(\mathbf{1};\varepsilon) \in [0,1)^d$ and that $\boldsymbol{g}'(x)$ is symmetric positive definite.*

Table 1: Comparison of potential thresholds with scalar recursion for LDPC based GT and BP threshold with coupling for $\Omega = 5\%$

	$d_{\mathsf{v}} = 3$	$d_{\mathsf{v}} = 4$	$d_{\mathsf{v}} = 5$	$d_{\mathsf{v}} = 6$	$d_{\mathsf{v}} = 10$
γ^*	0.63	1.20	1.74	2.24	3.81
BP $m = 10$	0.55	0.85	1.03	1.13	1.28

First Approach: Fixing Ω with Scalar Recursion

Substituting (13) into (12) we obtain

$$x_1^{(\ell)} = 1 - \left(1 - (1-\gamma)\left(x_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}\right)^{d_{\mathsf{c}}-1}. \tag{20}$$

Substituting (11) into (20) we obtain the scalar recursion

$$\begin{aligned} x_1^{(\ell)} &= 1 - \left(1 - (1-\gamma)\left(1 - \left(1 - y_1^{(\ell-1)}\right)^{d_{\mathsf{c}}-1}\right)^{d_{\mathsf{v}}-1}\right)^{d_{\mathsf{c}}-1} \\ y_1^{(\ell)} &= \gamma\left(x_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}. \end{aligned} \tag{21}$$

It can be shown that the recursion is a scalar admissible system. But it can be noticed that the recursion tracks only the probabilities connected to defective VNs. This masks the fact that the decoder has to access the non-defective VNs as well. Table 1 shows the potential thresholds as well as the BP threshold for coupling memory $m = 10$. It can be seen that the potential thresholds are too good when compared to the one obtained to the actual system with high coupling memory. So this rearrangement of the equations, while mathematically possible, cannot be physically realized when coupling is considered as it would imply separating the defective items and non-defective items.

Second Approach: Fixing γ with Vector Recursion

If we now examine the case where the rate Ω is fixed and the proportion γ of defectives is varied we observe that (17) can be rewritten as

$$\mathbf{f}_\gamma(x_0, x_1; \gamma) = \left[(1-\gamma)\cdot x_0^{d_{\mathsf{v}}-1}, \quad \gamma \cdot x_1^{d_{\mathsf{v}}-1}\right] \tag{22}$$

$$\mathbf{g}_\gamma(y_0, y_1) = \left[1 - (1-y_1)^{d_{\mathsf{c}}-1}, \quad 1 - (1-y_0)^{d_{\mathsf{c}}-1}\right], \tag{23}$$

where the $\mathbf{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are denoted by the parameter of interest γ. It can be observed that $\mathbf{f}_\gamma(x_0, x_1; \gamma)$ cannot satisfy the conditions for a scalar admissible system since one part is increasing with γ while another part is decreasing with γ.

We could, however, estimate the potential threshold γ^* from the curve obtained by computing Ω_{th} for a system with variable rate as shown in Fig. 5. This is done by drawing a horizontal line from the Ω axis to the curve and taking the γ value at the point of intersection as the γ^*_{th}.

Third Approach: Fixing Ω with Vector Recursion

We first consider the scenario where γ is fixed and the rate, Ω is changed by changing d_{c} (for a given d_{v}). Since d_{c} can be varied from 1 (best) to ∞ (worst), we have to define a parameter $\varepsilon(d_{\mathsf{c}})$ such that $\varepsilon(1) = 0$ and $\varepsilon(\infty) = 1$. The function

$$\varepsilon(d_{\mathsf{c}}) = 1 - \frac{1}{d_{\mathsf{c}}}$$

satisfies these requirements.

We can rewrite the DE equations (12)–(14) as

$$\mathbf{f}(y_0, y_1; d_{\mathsf{c}}) = \left[1 - (1 - y_1)^{d_{\mathsf{c}}-1}, \quad 1 - (1 - y_0)^{d_{\mathsf{c}}-1}\right] \tag{16}$$

$$\boldsymbol{g}(x_0, x_1) = \left[(1-\gamma) \cdot x_0^{d_{\mathsf{v}}-1}, \quad \gamma \cdot x_1^{d_{\mathsf{v}}-1}\right] . \tag{17}$$

Note that $\mathbf{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ have been changed compared to the second approach to reflect the parameter of interest which is now d_{c}. Substituting ε into (16) we obtain

$$\mathbf{f}(y_0, y_1; \varepsilon) = \left[1 - (1 - y_1)^{\frac{\varepsilon}{1-\varepsilon}}, \quad 1 - (1 - y_0)^{\frac{\varepsilon}{1-\varepsilon}}\right] \tag{18}$$

$$\boldsymbol{g}(x_0, x_1) = \left[(1-\gamma) \cdot x_0^{d_{\mathsf{v}}-1}, \quad \gamma \cdot x_1^{d_{\mathsf{v}}-1}\right] . \tag{19}$$

It can be shown that $\mathbf{f}(y_0, y_1; \varepsilon)$ is monotonically increasing with $\boldsymbol{y}$ and ε for $\varepsilon \in [0, 1]$ and $\mathbf{f}(0, 0; \varepsilon) = \mathbf{f}(y_0, y_1; 0) = \mathbf{0}$. Furthermore,

$$\boldsymbol{g}'(\boldsymbol{x}) = (d_{\mathsf{v}} - 1) \begin{bmatrix} (1-\gamma)x_0^{d_{\mathsf{v}}-2} & 0 \\ 0 & \gamma x_1^{d_{\mathsf{v}}-2} \end{bmatrix}$$

thus

$$|\boldsymbol{g}'(\boldsymbol{x})| = (d_{\mathsf{v}} - 1)\gamma(1-\gamma)(x_0 x_1)^{d_{\mathsf{v}}-2} > 0$$

for $x_0, x_1 > 0$. This implies that $\boldsymbol{g}'(x)$ is positive definite. We thus have a vector admissible system. We can then use the equation [8]

$$\mathrm{U}(\boldsymbol{x}; \varepsilon) = \int_0^{\boldsymbol{x}} \Big(\big(\boldsymbol{z} - \mathbf{f}(\boldsymbol{g}(\boldsymbol{z}); \varepsilon)\big) \boldsymbol{D}\boldsymbol{g}'(\boldsymbol{z}) \Big) \cdot \mathrm{d}\boldsymbol{z} \,, \tag{20}$$

to evaluate the potential threshold ε^*, defined as

$$\varepsilon^* = \sup\{\varepsilon \in [0, 1] \mid \min_{\boldsymbol{x}} \mathrm{U}(\boldsymbol{x}; \varepsilon) \geq 0\} \,. \tag{21}$$

$\boldsymbol{D}$ is a positive diagonal matrix which we set to the identity matrix in this case. The line integral for computing $\mathrm{U}(\boldsymbol{x}; \varepsilon)$ in (20) is path independent [8]. We can thus choose to compute the integral along a straight line in the direction defined by the vector from origin to $\boldsymbol{x}$. That is, we have $\boldsymbol{z}$ parameterized by λ as $\boldsymbol{z}(\lambda)$ and

$$\boldsymbol{z}(\lambda) = \lambda \boldsymbol{x} \implies z_1(\lambda) = \lambda x_1, \quad z_2(\lambda) = \lambda x_2 \,.$$

$$\mathrm{U}(\boldsymbol{x}; \varepsilon) = (1-\gamma)x_1^{d_{\mathsf{v}}-1} \left((1-\varepsilon) \frac{1 - \left(1 - \gamma x_2^{d_{\mathsf{v}}-1}\right)^{\frac{1}{1-\varepsilon}}}{\gamma x_2^{d_{\mathsf{v}}-1}} + \frac{(d_{\mathsf{v}}-1)}{d_{\mathsf{v}}} x_1 - 1 \right) + \gamma x_2^{d_{\mathsf{v}}-1} \left((1-\varepsilon) \frac{1 - \left(1 - (1-\gamma) x_2^{d_{\mathsf{v}}-1}\right)^{\frac{1}{1-\varepsilon}}}{(1-\gamma) x_1^{d_{\mathsf{v}}-1}} + \frac{(d_{\mathsf{v}}-1)}{d_{\mathsf{v}}} x_2 - 1 \right) \tag{23}$$

Hence we can write the potential function as [8]

$$\mathrm{U}(\boldsymbol{x};\varepsilon) = \int_0^1 \Big(\big(\boldsymbol{z}(\lambda) - \mathbf{f}(\boldsymbol{g}(\boldsymbol{z}(\lambda));\varepsilon)\big)\boldsymbol{D}\boldsymbol{g}'(\boldsymbol{z}(\lambda)) \Big) \cdot \boldsymbol{z}'(\lambda)\mathrm{d}\lambda\,. \tag{22}$$

The integral can be evaluated in closed form and is given in (23). Fig. 4 shows a 3 dimensional plot of the potential function for $d_{\mathsf{v}} = 6$ with $\gamma = 1\%$. To compute ε^*, x_1 and x_2 are each incremented by a small number Δ from 0 to 1 thus forming a two dimension grid. The value of U is computed for each point in the grid followed by evaluation of the minimum. This is done for each value of ε. The potential threshold is then determined using (21).

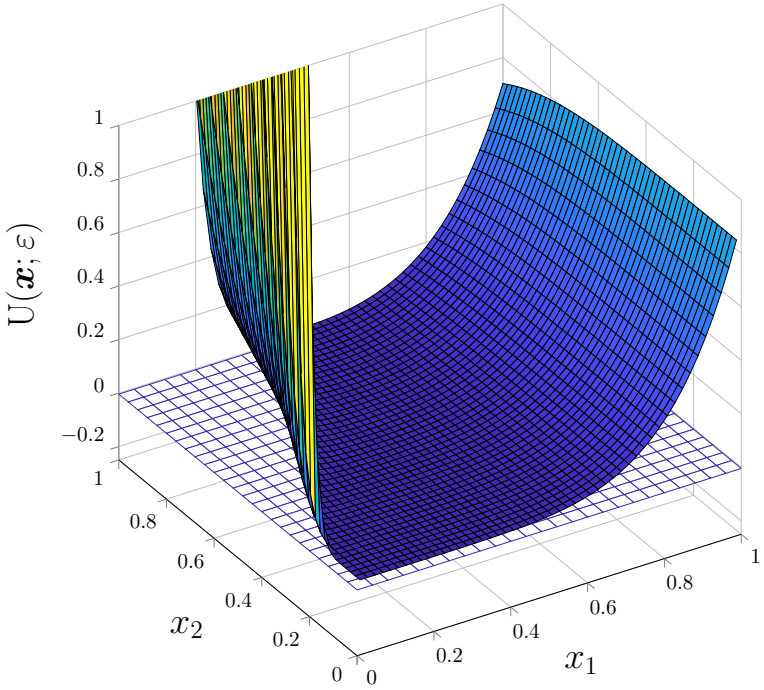


Figure 4: A 3D plot of the potential function $\mathrm{U}(\boldsymbol{x};\varepsilon)$ for $d_{\mathsf{v}} = 6$, $\gamma = 1\%$ with $\varepsilon^* = 0.9924$. The z axis is limited to 1 (values lager than 1 are not shown). $\mathrm{U}(\boldsymbol{x};\varepsilon)$ is above the $z = 0$ plane since $\varepsilon = 0.9667 < \varepsilon^*$.

The coupled system is guaranteed to converge to the zero point for all $\varepsilon < \varepsilon^*$. The minimum rate Ω_{th}, required for a coupled system can then be computed from ε^* as

$$\Omega^*_{\mathsf{th}} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} = d_{\mathsf{v}}(1-\varepsilon^*)\,.$$

An upper bound on ε^* corresponds to an upper bound on d_{c}, which in turn gives a lower bound on Ω_{th}[2]. Fig. 5 shows the results of a plot of Ω^*_{th} versus γ for various values of d_{v}. The results show that Ω^*_{th} improves (gets smaller) with increasing d_{v}. We see a tendency to converge for higher values of d_{v}. We have not

[2]It can be noted that only integer values of d_{c} are admissible for the regular graphs. If the value of Ω implies a non-integer d_{c}, we take the closest lower value of d_{c} which corresponds to a slightly higher Ω.

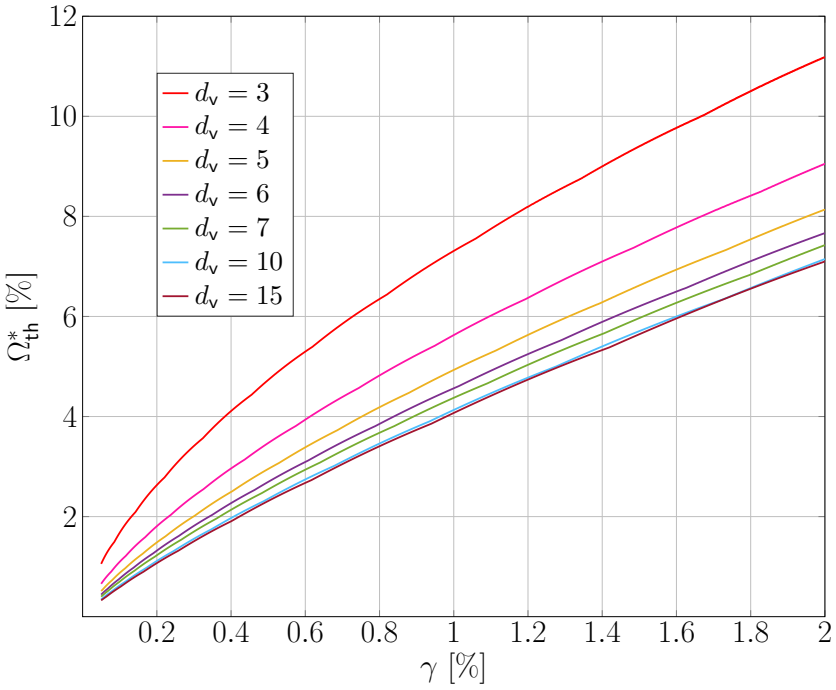


Figure 5: The minimum rate Ω^*_{th} for a fixed γ computed from the potential threshold ε^*.

however, been able to ascertain whether this corresponds to some fundamental limit.

Table 2 shows the BP thresholds of LDPC based GT for different values of d_{v} for fixed $\gamma = 1$ and $L = 5000$ with increasing coupling memory. The results are computed using the recursions in (11)-(14) for the uncoupled case and (16)-(19) for the coupled case. The value of d_{c} is increased from some small value (high rate) until the recursions are not able to converge to the zero point. It can be seen that for each d_{v} the threshold approaches the potential threshold Ω^*_{th} as predicted. Furthermore without coupling, the order of performance with d_{v} is not consistent but with coupling the performance is consistent with higher degrees showing better performance (lower rate).

On the other hand, Table 3 shows the DE results for a fixed rate $\Omega = 5\%$ and $L = 5000$. We use the same DE equations but fix d_{v} and d_{c} (thus fixing Ω) and increase γ. The results show the same trend as in the fixed γ case where the performance is less consistent with d_{v} without coupling but a more consistent improvement with increasing d_{v} when coupling is applied.

Fig. 6 shows the results of simulations with finite block length for Scenario 1 (fixed γ) for both the LDPC and GLDPC code based GT. The block size for the uncoupled and coupled case are both equal to 10^5. For the coupled case, we consider $w = 5$, $L = 200$ and the decoding is done on the full graph (not using a window decoder). It can be seen that even without coupling the LDPC scheme outperforms the GLDPC scheme and coupling widens the gap between the two more. As predicted by the threshold analysis, the performance of $d_{\mathsf{v}} = 10$ is poorer than $d_{\mathsf{v}} = 6$ without coupling, but this changes with coupling where

Table 2: Ω^*_{th} in % for $\gamma = 1\%$ with LDPC Code-Based Group Testing

d_{v}	$w=0$	$w=1$	$w=2$	$w=5$	$w=10$	$\Omega^*_{\mathsf{th}}[\%]$
4	7.27	5.71	5.63	5.63	5.63	5.63
5	6.94	5.15	4.95	4.95	4.95	4.95
6	7.06	5.00	4.65	4.58	4.58	4.58
7	7.22	5.04	4.49	4.40	4.40	4.38
8	7.41	5.13	4.47	4.26	4.26	4.26
9	7.63	5.23	4.48	4.17	4.17	4.17
10	7.87	5.38	4.57	4.13	4.13	4.13
15	9.32	6.20	5.10	4.14	4.08	4.08
20	10.70	7.02	5.71	4.38	4.07	4.07

Table 3: γ^*_{th} for $\Omega = 5\%$ with LDPC Code-Based Group Testing

d_{v}	$w=0$	$w=1$	$w=2$	$w=5$	$w=10$	$\gamma^*_{\mathsf{th}}[\%]$
4	0.60	0.84	0.85	0.85	0.85	0.85
5	0.64	0.96	1.02	1.03	1.02	1.02
6	0.65	1.00	1.11	1.13	1.13	1.13
7	0.64	1.00	1.16	1.19	1.19	1.19
8	0.62	0.98	1.16	1.23	1.23	1.23
9	0.60	0.95	1.15	1.26	1.26	1.26
10	0.58	0.92	1.13	1.27	1.28	1.28
15	0.49	0.78	0.98	1.26	1.29	1.29
20	0.42	0.67	0.86	1.18	1.26	1.29

$d_{\mathsf{v}} = 10$ becomes better. In this example, it can also be observed that for the GLDPC scheme, the gain with coupling is smaller than for the LDPC scheme, but the error floor is lowered significantly.

In the next part of the paper we explore ways to improve quantitative GT base on GLDPC codes.

5 Improving Quantitative GT Based on GLDPC Codes

As it has been observed in Section 3 the LDPC based quantitative GT outperforms the GLDPC based scheme. In this section, we highlight some of the reasons for the inferior performance of the GLDPC based scheme and show what can be done to improve its performance. Three improvements are considered in this section for the GLDPC based scheme. First we formally introduce coupling, prove threshold saturation and compare the potential thresholds with those of the LDPC counterpart. Then we propose two ways to improve the decoder for better performance.

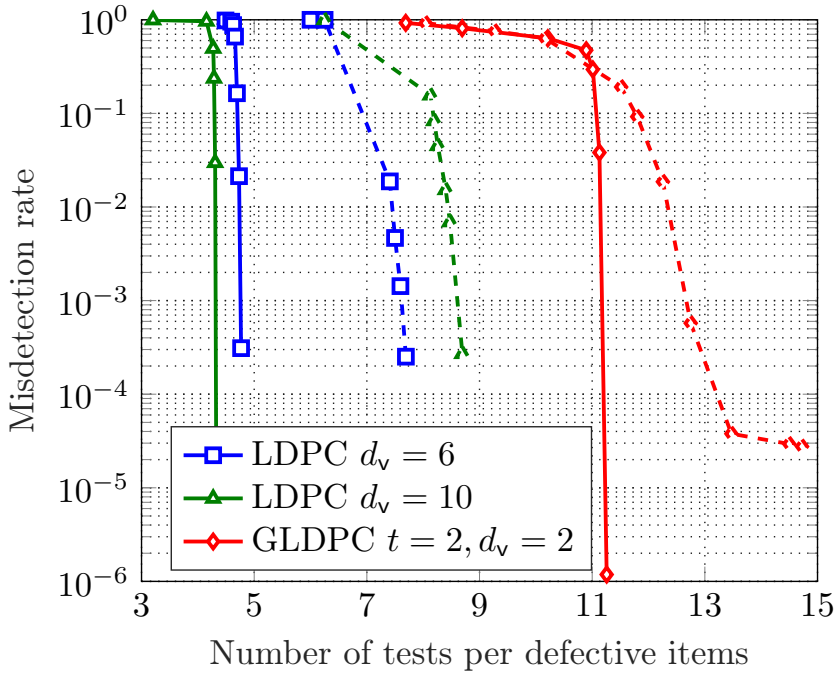


Figure 6: Misdetection rate as a function of number of tests per defective items for a fixed $\gamma = 1\%$ for uncoupled (dashed) and coupled (solid) GT scheme.

5.1 Threshold Saturation for GLDPC Based GT

For GLDPC codes coupling is done on the adjacency matrix $\boldsymbol{B}$. We thus have a similar procedure as for the LDPC described in Section 4 with the exception that the CNs now correspond to rows of $\boldsymbol{B}$ instead of $\boldsymbol{A}$. The lower degrees at the boundaries increase the likelihood of having tests with t or fewer defective items, leading to better resolving of items.

Here we prove threshold saturation for the GLDPC based quantitative GT using the decoder proposed in [5] whose DE equations are given in (5)-(6). We consider the scenario where the rate Ω is fixed and γ is varied. We can readily obtain a *scalar admissible system* by defining $x^{(\ell)} = 1 - q^{(\ell)}$, which represents the probability that a CN send a message *unresolved* to a defective VN. We thus have the recursion as

$$g(x) = 1 - \sum_{i=0}^{t-1} \binom{d_c - 1}{i} x^i \, (1-x)^{d_c - 1 - i} \tag{24}$$

$$f(y) = \gamma \, y^{d_v - 1} \,, \tag{25}$$

It can readily be seen that $f(y)$ is strictly increasing in both y and γ. For $g(x)$ we have the negative term as a binomial cdf which can be shown to be decreasing with x, implying $g(x)$ is increasing with x. Hence the recursion is a scalar admissible system. The potential function is then given as [16]

$$U(x;\gamma) = x g(x) - G(x) - F(g(x);\gamma) \,,$$

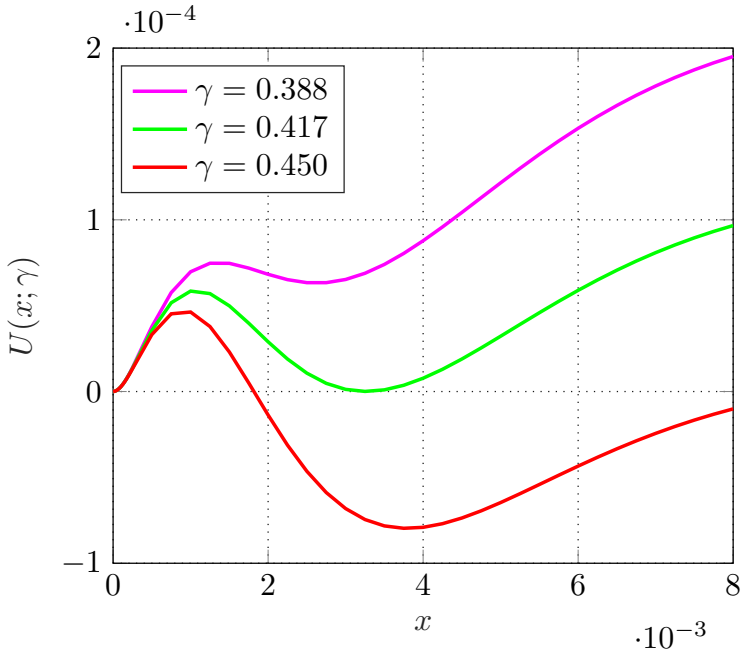


Figure 7: A plot of the potential function $U(x;\gamma)$ for GLDPC based group testing with BMD decoder for $t = 1$ and $d_{\mathsf{v}} = 3$ and $\Omega = 5\%$ for different values of γ. It can be observed that $\gamma^*_{\mathsf{th}} = 0.417$.

where $F(x) = \int_0^x f(z)dz = \frac{\gamma}{d_{\mathsf{v}}} x^{d_{\mathsf{v}}}$ and $G(x) = \int_0^x g(z)dz$. We cannot easily have a general closed form for $G(x)$ but we can compute it for given values of d_{c} and t. Fig. 7 shows the potential function for $t = 1$, $d_{\mathsf{v}} = 3$. The potential threshold for GLDPC based GT is plotted in Fig.8 for a range of Ωs. For comparison, the potential thresholds of the LDPC based scheme are also plotted in the same figure. It can be seen that potential thresholds of the GLDPC scheme is worse than the LDPC scheme with the gap increasing with increasing γ. The gap between GLDPC and LDPC also increases with coupling which implies the GLDPC scheme benefits less from spatial coupling.

5.2 Improving the GLDPC Decoder using Non-defective Items

Since the decoder uses the t-error correction capability of BCH codes whereby the defective items are considered as the "errors" we can improve the decoder by also considering the non-defective items as the "errors". This has almost no effect at the start of the decoding iterations since almost all tests have more than t non-defective items. As the peeling progresses we can then encounter the situation whereby the number of defective items might be more than t but the number of non-defective items is less than t. To decode for non-defective items the following is done. For each individual test i in $\boldsymbol{A}$ with degree $d^{(\ell)}_{\mathsf{c,i}}$ we define a complement syndrome s^{c}_i as

$$s^{\mathsf{c}}_i = d^{(\ell)}_{\mathsf{c,i}} - s^{(\ell)}_i \text{ for } i = 1, 2, \ldots, m\,. \tag{26}$$

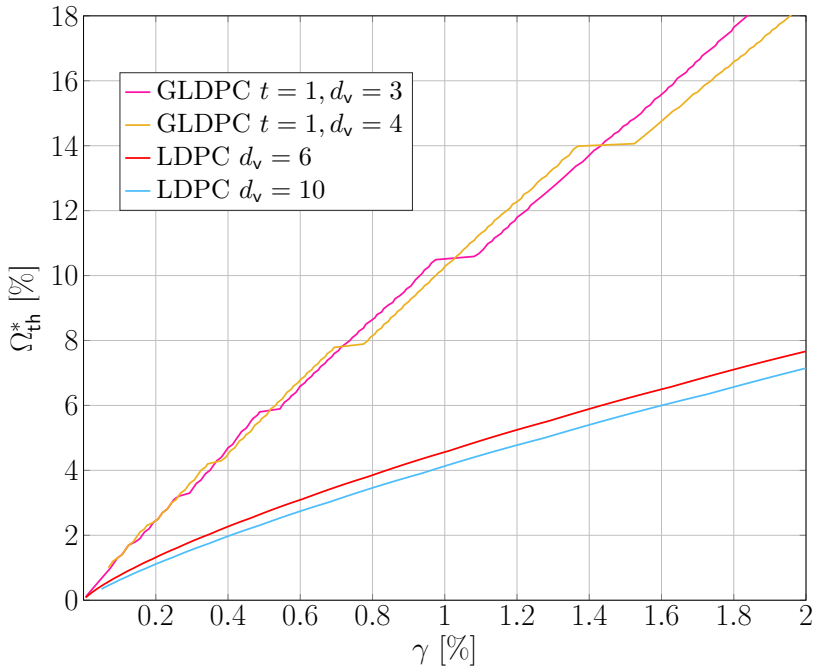


Figure 8: The potential thresholds for the GLDPC scheme for $t = 1$. The potential thresholds for the LDPC scheme are also shown for comparison. It can be seen that the LDPC scheme is better with the gap increasing with γ

In other words we define the new syndrome as the count of non-defective items. It can be noted that this new syndrome is obatined without additional tests but by simply manipulating the original test results using equation (26). We can then apply the same decoder as for the defective items discussed above. For convenience we refer the decoder identifying positions of defectives as the "ones-decoder" while the one identifying positions of non-defectives as the "zeros-decoder".

It can be observed that when a BCH decoder has identified the "error" positions, these positions are given the label 1 while it infers the remaining items connected to a CN in the pruned graph to be zero. This means that non-defective items can be resolved by the 'ones-decoder' as well. The same applies to the 'zeros-decoder' that it also labels the remaining items as ones. Notice that these labels affect the unresolved items only since the pruned graph has does not contain any resolved items thus ruling out any label oscillations. We refer to the decoder which uses both the ones and zeros as the double-sided decoder (DSD) and the conventional decoder as the single-sided decoder (SSD).

The DE equations (5)-(6) have to be modified to include the probabilities for non-defective items. Let $q_0^{(\ell)}$ and $q_1^{(\ell)}$ be the probabilities that a CN sends a message *resolved* to a non-defective and defective VN, respectively, during iteration ℓ. Also let $p_0^{(\ell)}$ denote the probability that an item is non-defective and sends a message *unresolved* to a CN, and $p_1^{(\ell)}$ the probability that an item is defective and sends a message *unresolved* to a CN.

Proposition 5. *The quantities $q_0^{(\ell)}$, $q_1^{(\ell)}$, $p_0^{(\ell)}$, and $p_1^{(\ell)}$ are given by the following*

DE equations

$$
\begin{aligned}
q_1^{(\ell)} &= \sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_1^{(\ell-1)}\right) + \\
&\quad \left(1-\sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_1^{(\ell-1)}\right)\right) \sum_{i=0}^{t} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_0^{(\ell-1)}\right) , \\
q_0^{(\ell)} &= \sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_0^{(\ell-1)}\right) + \\
&\quad \left(1-\sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_0^{(\ell-1)}\right)\right) \sum_{i=0}^{t} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_1^{(\ell-1)}\right) , \\
p_1^{(\ell)} &= \gamma\left(1-q_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} , \\
p_0^{(\ell)} &= (1-\gamma)\left(1-q_0^{(\ell-1)}\right)^{d_{\mathsf{v}}-1} .
\end{aligned}
$$

Proof. The DE equations are derived using the assumption that the two decoders access the same updated value from each of the VNs. A message from a CN to a VN corresponding to a defective item is *unresolved* if both the 'ones-decoder' and the 'zeros-decoder' cannot resolve it. The 'ones-decoder' cannot resolve a VN corresponding to a defective item if more than $t-1$ of the other $d_{\mathsf{c}}-1$ VNs connected to the CN are defective and sends the message *unresolved* in the previous iteration. This happens with probability $1-\sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_1^{(\ell-1)}\right)$. The 'zeros-decoder' cannot resolve a defective VN if more than t of the other VNs are non-defective and unresolved which happens with probability $1-\sum_{i=0}^{t} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_0^{(\ell-1)}\right)$. The probability of at least one of the two decoders succeeding is given by

$$
1-\left(1-\sum_{i=0}^{t-1} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_1^{(\ell-1)}\right)\right)\left(1-\sum_{i=0}^{t} \mathsf{BN}\left(d_{\mathsf{c}}-1, i, p_0^{(\ell-1)}\right)\right) .
$$

Expanding this gives $q_1^{(\ell)}$. We can use the same arguments to obtain $q_0^{(\ell)}$. The message from a defective VN to a CN is unresolved if all the other $d_{\mathsf{v}}-1$ CNs send the message *unresolved*. Thus the probability that an item is defective and sends the message *unresolved* is given by $p_1^{(\ell)} = \gamma\left(1-q_1^{(\ell-1)}\right)^{d_{\mathsf{v}}-1}$. Similar reasoning gives $p_0^{(\ell)}$ for the non-defective VN. ■

Table 4 shows thresholds in terms of maximum γ which can be resolved with vanishing error probability for different values of Ω for both the SSD and the DSD. It can be seen that for $t=1$ and $d_{\mathsf{v}}=2$ the DSD has better thresholds for all rates. But with higher values of d_{v} the two decoders cannot be distinguished for the accuracy we used except for Ω as high as 35%. For $t=2$ and higher (not shown in this paper) the two decoders do not exhibit any noticeable difference in thresholds. The real effect of the DSD is observed with finite length simulations as shown in Fig. 9. The DSD has the error floor significantly reduced compared to the SS decoder.

Table 4: Comparison of the GLDPC with and without DSS decoder different fixed values of Ω showing maximum γ

t	d_v	$\Omega = 5\%$		$\Omega = 25\%$		$\Omega = 35\%$	
		SS	DSD	SSD	DSD	SSD	DSD
1	2	0.25	0.27	1.81	2.39	2.56	3.56
	3	0.37	0.37	2.56	2.56	4.11	4.13
	4	0.35	0.35	2.14	2.14	3.39	3.39
2	2	0.39	0.39	2.80	2.80	3.97	3.97
	3	0.33	0.33	2.28	2.28	3.20	3.20
	4	0.28	0.28	1.75	1.75	2.74	2.74

5.3 Improving the GLDPC Decoder using the LDPC Decoder

One would have expected that increasing the error correcting capability t of the component BCH code would improve the performance but this is not the case. If we look at (3), we see that the rate increases almost linearly with t. This increase in the number of tests is not compensated by the increase in performance. The LDPC scheme can be seen as having $t = 0$ thus eliminating the term with t. But even if the parameters are adjusted such that we have a fixed rate, the performance is inferior to the LDPC based scheme.

One reason for this performance gap lies in the decoder. First, to decode the test results using the algebraic decoder of BCH codes, we need the test results except the all ones row to be reduced modulo 2 (to either 0 or1). This results in a loss of information. Secondly, the BMD decoder is not an optimum decoder even if we were to consider only hard information. This is because there are error patterns with weight more than t which could potentially be corrected. The combination of these two weaknesses makes the decoder sub-optimal.

It can be observed that the component code $\boldsymbol{H}$ of the GLDPC scheme is in fact an irregular LDPC code. Each component code can be decoded using the decoding algorithm for the LDPC based GT as described in Section 3.2. We can thus improve the GLDPC performance by utilizing the LDPC decoder. The algorithm works as follows.

1. The BMD decoder resolves all tests with t or fewer defective items
2. The LDPC decoder is run on each component code for those tests with more than t defective/non-defective items
3. If any of the steps $(1) - (2)$ has resolved any item the algorithm restarts step (1) otherwise the algorithm halts.

We refer to this modified decoder as the low-density decoder (LDD). This new decoder results in a significant improvement in the waterfall performance of the GLDPC based GT. Fig. 9 shows the effect of this improvement when in

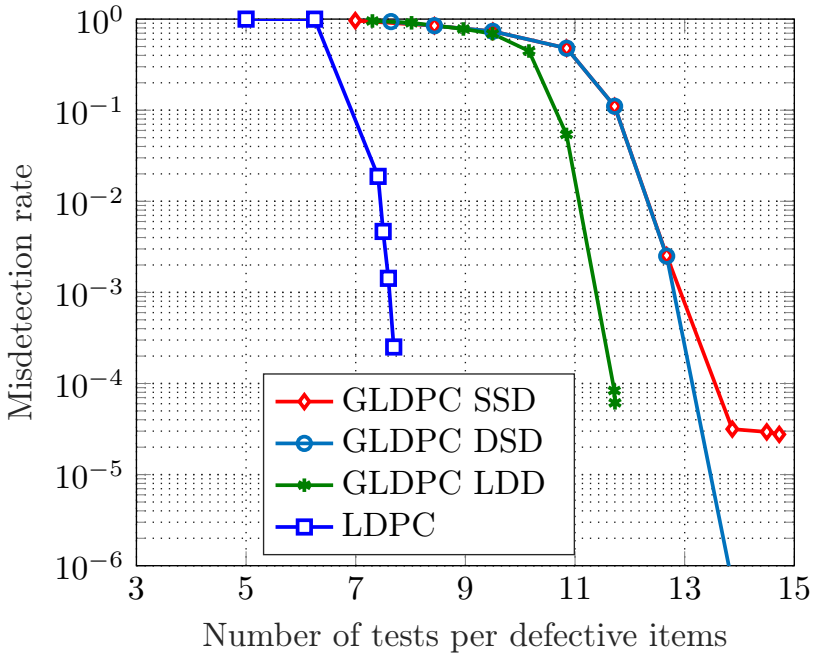


Figure 9: Comparison of the GLDPC ($t = 2, d_{\mathsf{v}} = 2$) with the SSD in [5], the DSD as well as the LDD for $\gamma = 1\%$. Also shown is the LDPC with $d_{\mathsf{v}} = 6$.

comparison to the original BMD decoder in [5] and the LDPC based scheme.

References

[1] P. Damaschke, "Threshold group testing" in General Theory of Information Transfer and Combinatorics (Lecture Notes in Computer Science), Berlin, Germany:Springer-Verlag, vol. 4123, pp. 707-718, 2006.

[2] A. Emad and O. Milenkovic, "Semiquantitative group testing," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4614–4636, 2014.

[3] K. Lee, R. Pedarsani, and K. Ramchandran, "SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July 2016.

[4] A. Vem, N. T. Janakiraman, and K. R. Narayanan, "Group testing using left-and-right-regular sparse-graph codes," *CoRR*, vol. abs/1701.07477, 2017. [Online]. Available: http://arxiv.org/abs/1701.07477

[5] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *Proc. IEEE Inf. Theory Work. (ITW)*, 2019.

[6] ——, "Non-adaptive quantitative group testing using irregular sparse graph codes," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019.

[7] M. M. Mashauri, A. Graell i Amat, and M. Lentmaier, "Low-density parity-check codes and spatial coupling for quantitative group testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, June 2023.

[8] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions," in *Proc. IEEE Inf. Theory Work. (ITW)*, Lausanne, Switzerland, Sep. 2012.

[9] A. G. Dyachkov and V. V. Rykov, "A coding model for a multiple-access adder channel." *Problems of Information Transmission*, vol. 17, no. 2, pp. 26–38, 1981.

[10] B. S. Tsybakov, "Resolution of a conflict of known multiplicity," *Problems of Information Transmission*, vol. 16, no. 2, pp. 69–82, 1980.

[11] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[12] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications 2nd Ed.*, ser. Pearson education. Pearson-Prentice Hall, 2004.

[13] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

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

[15] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE. Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, 2015.

[16] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions," in *Proc. IEEE Int. Symp. on Turbo Codes and Iterative Info. Processing (ISTC)*, Gothenburg, Sweden, Oct. 2012

Paper VI

LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet

We propose and analyze a novel scheme based on LDPC codes for quantitative group testing. The key underlying idea is to augment the bipartite graph by introducing hidden non-binary variables to strengthen the message-passing decoder. This is achieved by grouping items into bundles of size q within the test matrix, while keeping the testing procedure unaffected. The decoder, inspired by some works on counter braids, passes lower and upper bounds on the bundle values along the edges of the graph, with the gap between the two shrinking with the decoder iterations. Through a density evolution analysis and finite length simulations, we show that the proposed scheme significantly outperforms its binary counterpart with limited increase in complexity.

Keywords: Quantitative Group testing, Low density parity check codes, Counter braids

---

Mgeni Makambi Mashauri, Alexandre Graell i Amat, and Michael Lentmaier, 'LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet", *Submitted to IEEE Communication Letters in June 2024 (under review).*

1 Introduction

Group testing (GT) finds application across a variety of fields, including medicine, data forensics, and communications [1–3]. Sparse codes-on-graphs with message passing decoding have recently been shown to be efficient for GT when the fraction of defective items is very small [4, 5]. In multi-access communications, for example, the goal is to accommodate many devices in a network with limited time and spectral resources. In massive machine-type communication (MMTC), a network has to handle traffic from a large population of sensors and other intelligent devices [6]. Only a few of these devices are active at any particular time. A network with time slots is an instance of a GT problem with the slots as tests and the devices as items. This has been investigated in various works [3, 7, 8]. Quantitative GT captures the model known as the adder channel [9], as well as the so-called collision with known multiplicity [10, 11].

Karimi *et al.* [4, 5] proposed a quantitative GT scheme based on generalized LDPC (GLDPC) codes with t-error correcting BCH codes as component codes. Following up on this scheme, we discovered in [12] that GT based on simple LDPC codes, with $t = 0$ and a peeling decoder of lower complexity, is more effective in reducing the number of tests. In the same work, it was shown that spatial coupling improves both the LDPC and GLDPC scheme.

In this work, we introduce q-bundles of items as a method to further improve the uncoupled LDPC code scheme in [12] through a non-binary alphabet and a corresponding novel decoder that is inspired by counter braids [13, 14]. A few extra conventional tests without bundles are added to resolve individual items from the estimated bundles, reducing the overall number of required tests compared to the original scheme in [12]. Furthermore, we derive density evolution equations for the proposed decoder and compute the corresponding decoding thresholds. Finite-length simulations for a large population size are presented and compared to the asymptotic thresholds.

2 System Model

We consider a population of n items represented by a binary vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i = 1$ if item i is defective and $x_i = 0$ if it is not. Each item is defective with probability γ.

The GT scheme aims at recovering $\boldsymbol{x}$ using m tests, where $m < n$, and can be represented by an $m \times n$ adjacency matrix $\boldsymbol{A} = (a_{i,j})$, whereby $a_{i,j} = 1$ if item j participates in test i and $a_{i,j} = 0$ otherwise. We consider noiseless quantitative GT whereby the result of each test indicates the exact number of defective items participating in the test. The result of all tests can be collected in a vector $\boldsymbol{s} = (s_1, \ldots, s_m)$, referred to as the syndrome, where s_i is the result of test i. We thus have

$$s_i = \sum_{j=1}^{n} x_j a_{i,j} \quad \text{and} \quad \boldsymbol{s} = \boldsymbol{x}\boldsymbol{A}_{\mathrm{T}}\,.$$

The test assignment can alternatively be visualized through a bipartite graph corresponding to matrix $\boldsymbol{A}$ with n variable nodes (VNs) representing the items and m constraint nodes (CNs) representing the tests. In this work, we consider

a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ bipartite graph whereby each VN is connected to d_{v} CNs and each CN is connected to d_{c} VNs.

3 Proposed Non-Binary Group Testing Scheme

We propose a GT scheme whereby items are grouped into bundles of size q. From the bipartite graph perspective, this corresponds to introducing non-binary variables, which take values in $[0, q]$. The underlying idea is that non-binary variables strengthen the message-passing decoder, leading to improved performance as we shall see in Sections 5 and 6. The grouping puts restrictions on the test assignment matrix $\boldsymbol{A}$ and the corresponding graph. We describe the construction of the graph in the following.

We augment the conventional bipartite graph for GT by introducing additional n_{h} *hidden* VNs and n_{h} CNs corresponding to bundles of q items. The m tests are then divided into two sets, $\mathcal{CN}_z$ and $\mathcal{CN}_x$, of cardinality m_z and m_x, respectively. Tests in $\mathcal{CN}_z$ are connected to hidden VNs, while tests in $\mathcal{CN}_x$ are connected to VNs corresponding to items. We refer to VNs corresponding to items as *conventional* VNs. We denote by d_{v} and $d_{\mathsf{v,x}}$ the degree and lower-degree of conventional VNs; each conventional VN is connected to d_{v} CNs in $\mathcal{CN}_x \cup \mathcal{CN}_z$ and to $d_{\mathsf{v,x}}$ CNs in $\mathcal{CN}_x$. Furthermore, we denote by $d_{\mathsf{v,z}}$ the degree of hidden VNs; each hidden VN is connected to $d_{\mathsf{v,z}}$ CNs in $\mathcal{CN}_z$. Note that $d_{\mathsf{v}} = d_{\mathsf{v,x}} + d_{\mathsf{v,z}}$. Similarly, we denote by d_{c} and $d_{\mathsf{c,z}}$ the degree of CNs in $\mathcal{CN}_x$ and $\mathcal{CN}_z$, respectively. In correspondence to the tests in $\mathcal{CN}_z$ and $\mathcal{CN}_x$, the syndrome $\boldsymbol{s}$ is also split into two parts, $s_q(\mathbf{c})$ and $s_x(\mathbf{c})$.

Fig. 1 shows the augmented graph corresponding to a GT scheme with $n = 8$ items, $m = 6$ tests, and 4 bundles of $q = 2$ items each. The bundles are represented by the 4 hidden VNs labeled $z_1, \ldots, z_4$ and the 4 hidden CNs labeled $\mathsf{f}_1, \ldots, \mathsf{f}_4$. Here, $d_{\mathsf{v}} = 3$, $d_{\mathsf{v,x}} = 1$, and $d_{\mathsf{v,z}} = 2$. The corresponding adjacency matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \end{bmatrix}.$$

It can be seen from the graph that $z = \mathsf{f}(\boldsymbol{x}) = \sum_{i:x_i \in \mathcal{N}_x(\mathsf{f})} x_i$, where $\mathcal{N}_x(\mathsf{f})$ is the set of items grouped in bundle f. Compared to a conventional bipartite graph, we have thus introduced additional hidden CNs f and VNs z.

The number of edges adjacent to conventional VNs must be equal to number of edges adjacent to CNs in $\mathcal{CN}_x$. Hence, $m_x d_{\mathsf{c}} = n d_{\mathsf{v,x}}$. Furthermore, the number of edges adjacent to hidden VNs must be equal to number of edges adjacent to CNs in $\mathcal{CN}_z$. Thus, $m_z d_{\mathsf{c,z}} = n_{\mathsf{h}} d_{\mathsf{v,z}}$. We set $d_{\mathsf{c,z}} = d_{\mathsf{c}}/q$, so that all CNs are connected to d_{c} conventional VNs (in the case of CNs in $\mathcal{CN}_z$, via the corresponding hidden VN).

It is important to highlight that the tests performed on the items are oblivious to the bundles. However, the decoder can take advantage of the bundles.

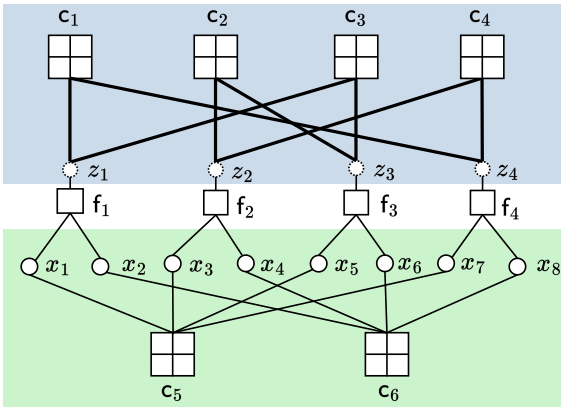


Figure 1: Graphical representation of a system with $q = 2$, $d_{\mathsf{c}} = 4$, $d_{\mathsf{v}} = 3$, $d_{\mathsf{v},\mathsf{x}} = 1$ and $d_{\mathsf{v},\mathsf{z}} = 2$. Tests are represented by square with a plus sign while empty squares represents bundles. In this case $\mathcal{CN}_z = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$ while $\mathcal{CN}_x = \{\mathsf{c}_5, \mathsf{c}_6\}$. All tests have the same degree $d_{\mathsf{c}} = 4$ since each edge from a bundle to a test corresponds to two edges in the overall graph.

4 Message Passing Decoder

In correspondence to the augmented graph in Fig. 1, there are three interactions in the message passing decoder: i) a test-bundle interaction, whereby CNs in $\mathcal{CN}_z$ pass messages to the hidden VNs corresponding to the bundles and vice-versa, ii) an item-bundle interaction, whereby conventional VNs pass messages to the hidden CNs representing the bundles and vice-versa, and iii) a test-item interaction, whereby CNs in $\mathcal{CN}_x$ and conventional VNs exchange messages. We use a scheduling wherein the messages are first passed from CNs in $\mathcal{CN}_z$ to hidden VNs, then from hidden VNs to hidden CNs, then from hidden CNs to conventional VNs, and finally from conventional VNs to CNs in $\mathcal{CN}_x$. This is then followed by the reverse, starting from CNs $\mathcal{CN}_x$ to conventional VNs and so on.

4.1 Test-Bundle Messages

In the test-bundle interaction, the optimal local decoder is a symbol-wise a-posterior-probability (APP) decoder that can be implemented in a trellis. The complexity of such a decoder, however, becomes infeasible with increasing check node degree $d_{\mathsf{c},\mathsf{z}}$. To reduce complexity, we use a hard decision decoder similar to the one used in [14] for counter braids (with some minor modifications). The simplification is achieved by neglecting the actual distribution of the value of a bundle and assigning a uniform distribution from some minimum value to a maximum value. This means that instead of passing a vector with $q + 1$ entries, the decoder passes lower and upper bounds only. For convenience, we use L-bound and U-bound for the lower and upper bounds, respectively.

The message passed from a CN $\mathsf{c} \in \mathcal{CN}_z$ to a hidden VN z at iteration ℓ is a pair of values $[\mathsf{L}_{\mathsf{c}\to z}^{(\ell)}, \mathrm{U}_{\mathsf{c}\to z}^{(\ell)}]$ given as

$$\mathsf{L}_{\mathsf{c}\to z}^{(\ell)} = \max\left\{ s(\mathsf{c}) - \sum_{z' \in \mathcal{T}(\mathsf{c})\setminus z} \mathrm{U}_{z'\to \mathsf{c}}^{(\ell-1)}, 0 \right\}, \tag{1}$$

$$\mathrm{U}_{\mathsf{c}\to z}^{(\ell)} = \min\left\{ s(\mathsf{c}) - \sum_{z'\in\mathcal{T}(\mathsf{c})\setminus z} \mathsf{L}_{z'\to\mathsf{c}}^{(\ell-1)}, q \right\}. \tag{2}$$

The L-bound $\mathsf{L}_{\mathsf{c}\to z}^{(\ell)}$ comes from the fact that the value of the syndrome corresponding to CN c is equal to the sum of the values of its neighboring hidden VNs. The L-bound is then obtained assuming all these values take on their maximum, i.e., the corresponding U-bound, and knowing that it cannot be negative. The U-bound $\mathrm{U}_{\mathsf{c}\to z}^{(\ell)}$ is obtained similarly.

The message from a hidden VN z to a CN $\mathsf{c}\in\mathcal{CN}_z$ is

$$\mathsf{L}_{z\to\mathsf{c}}^{(\ell)} = \max\left\{ \max_{\mathsf{c}'\in\mathcal{T}(z)\setminus\mathsf{c}} \mathsf{L}_{\mathsf{c}'\to z}^{(\ell-1)},\ \mathsf{L}_{\mathsf{f}\to z}^{(\ell)} \right\} \tag{3}$$

$$\mathrm{U}_{z\to\mathsf{c}}^{(\ell)} = \min\left\{ \min_{\mathsf{c}'\in\mathcal{T}(z)\setminus\mathsf{c}} \mathrm{U}_{\mathsf{c}'\to z}^{(\ell-1)},\ \mathrm{U}_{\mathsf{f}\to z}^{(\ell)} \right\}, \tag{4}$$

where $\mathsf{L}_{\mathsf{f}\to z}^{(\ell)}$ and $\mathrm{U}_{\mathsf{f}\to z}^{(\ell)}$ are given by

$$\mathsf{L}_{\mathsf{f}\to z}^{(\ell)} = \sum_{x\in\mathcal{N}(\mathsf{f})} \mathsf{L}_{x\to\mathsf{f}}^{(\ell)} \quad\text{and}\quad \mathrm{U}_{\mathsf{f}\to z}^{(\ell)} = \sum_{x\in\mathcal{N}(\mathsf{f})} \mathrm{U}_{x\to\mathsf{f}}^{(\ell)}. \tag{5}$$

Similarly, the message from a hidden VN z to a hidden CN f is

$$\mathsf{L}_{z\to\mathsf{f}}^{(\ell)} = \max_{\mathsf{c}\in\mathcal{T}(z)} \mathsf{L}_{\mathsf{c}\to z}^{(\ell)}, \qquad \mathrm{U}_{z\to\mathsf{f}}^{(\ell)} = \min_{\mathsf{c}\in\mathcal{T}(z)} \mathrm{U}_{\mathsf{c}\to z}^{(\ell)}. \tag{6}$$

4.2 Item-Bundle Messages

In the item-bundle interaction, the message passed from a hidden CN f to a conventional VN x is the pair of integers

$$\mathsf{L}_{\mathsf{f}\to x}^{(\ell)} = \max\left\{ \mathsf{L}_{z\to\mathsf{f}}^{(\ell-1)} - \sum_{x'\in\mathcal{N}(\mathsf{f})\setminus x} \mathrm{U}_{x'\to\mathsf{f}}^{(\ell-1)}, 0 \right\} \tag{7}$$

$$\mathrm{U}_{\mathsf{f}\to x}^{(\ell)} = \min\left\{ \mathrm{U}_{z\to\mathsf{f}}^{(\ell-1)} - \sum_{x'\in\mathcal{N}(\mathsf{f})\setminus x} \mathsf{L}_{x'\to\mathsf{f}}^{(\ell-1)}, 1 \right\}, \tag{8}$$

while the message passed from a conventional VN to a hidden CN is

$$\mathsf{L}_{x\to\mathsf{f}}^{(\ell)} = \max_{\mathsf{c}\in\mathcal{T}_s(x)} \mathsf{L}_{\mathsf{c}\to x}^{(\ell-1)} \quad\text{and}\quad \mathrm{U}_{x\to\mathsf{f}}^{(\ell)} = \min_{\mathsf{c}\in\mathcal{T}_s(x)} \mathrm{U}_{\mathsf{c}\to x}^{(\ell-1)}, \tag{9}$$

where $\mathsf{c}\in\mathcal{T}_s(x)$ is the set of CNs in $\mathcal{CN}_x$ connected to x.

4.3 Test-Item Messages

For the test-item interaction, the message exchange is similar to that of the test-bundle interaction with $q=1$. Note that, for each unresolved conventional VN, the L-bound is 0 and U-bound is 1. Furthermore, for a resolved conventional VN, the L-bound equals to the U-bound. Thus, the message from a CN $\mathsf{c}\in\mathcal{CN}_x$ to a conventional VN x is

$$\mathsf{L}_{\mathsf{c}\to x}^{(\ell)} = \max\left\{ s(\mathsf{c}) - \sum_{x'\in\mathcal{T}(\mathsf{c})\setminus x} \mathrm{U}_{x'\to\mathsf{c}}^{(\ell-1)}, 0 \right\} \tag{10}$$

$$\mathrm{U}_{\mathsf{c}\to x}^{(\ell)} = \min\left\{ s(\mathsf{c}) - \sum_{x'\in\mathcal{T}(\mathsf{c})\setminus x} L_{x'\to\mathsf{c}}^{(\ell-1)}, 1 \right\}, \tag{11}$$

and the message from a conventional VN x to a CN $\mathsf{c} \in \mathcal{CN}_x$ is

$$\mathsf{L}_{x\to\mathsf{c}}^{(\ell)} = \max\left\{\max_{\mathsf{c}'\in\mathcal{T}_s(x)\setminus\mathsf{c}} L_{\mathsf{c}'\to x}^{(\ell-1)}, \mathsf{L}_{\mathsf{f}\to x}^{(\ell-1)}\right\} \tag{12}$$

$$\mathrm{U}_{x\to\mathsf{c}}^{(\ell)} = \min\left\{\min_{\mathsf{c}'\in\mathcal{T}_s(x)\setminus\mathsf{c}} U_{\mathsf{c}'\to x}^{(\ell-1)}, \mathrm{U}_{\mathsf{f}\to x}^{(\ell-1)}\right\}. \tag{13}$$

It can be shown that the decoder for the test-item interaction is equivalent to the decoder in [12].

5 Density Evolution

In this section, we derive the density evolution equations for the decoder discussed above. For the test-bundle interaction, we have to determine the probability mass function (pmf) of the U-bound and L-bound on the value of the bundle. Let Z be the random variable corresponding to bundle $\boldsymbol{z}$. The U-bound on Z cannot be smaller than the value of Z and the L-bound cannot be greater than the value of Z. This means that the pmfs are implicitly conditioned on the value of Z.

Unless stated otherwise, we use the notation $\mathrm{P}_X\big(x\big) = \mathrm{P}\big(X = x\big)$ for the pmf and $\mathrm{F}_X\big(x\big)$ for the cdf.

5.1 Density Evolution for Test-Bundle Interaction

We first begin with the message passed from a CN $\mathsf{c} \in \mathcal{CN}_z$ to a hidden VN z. Making reference to (1), let $S_{\sim\boldsymbol{z}}^{u} = \sum_{z'\in\mathcal{T}(\mathsf{c})\setminus z} \mathrm{U}_{z'\to\mathsf{c}}^{(\ell-1)}$. The pmf of $S_{\sim\boldsymbol{z}}^{u}$ is obtained as the convolution of the individual conditional pmfs $\mathrm{P}_{\mathsf{U}_{z'\to\mathsf{c}}^{(\ell-1)}|Z}\big(y|z\big)$. The pmf for the L-bound is then

$$\mathrm{P}_{\mathsf{L}_{\mathsf{c}\to z}^{(\ell)}|Z}\big(i|z\big) = \begin{cases} \big(1 - \mathrm{F}_{S_{\sim\boldsymbol{z}}^{u}}\big(s(\mathsf{c})-1\big)\big) & i = 0 \\ \mathrm{P}_{S_{\sim\boldsymbol{z}}^{u}}\big(s(\mathsf{c})-i\big) & 0 < i \le z \\ 0 & \text{otherwise}. \end{cases} \tag{14}$$

Similarly, referring to (2) for the U-bound and letting $S_{\sim z}^{l} = \sum_{z'\in\mathcal{T}(\mathsf{c})\setminus z} \mathsf{L}_{z'\to\mathsf{c}}^{(\ell-1)}$, we have

$$\mathrm{P}_{\mathrm{U}_{\mathsf{c}\to z}^{(\ell)}|Z}\big(i|z\big) = \begin{cases} \mathrm{F}_{S_{\sim z}^{l}}\big(s(\mathsf{c})-q\big) & i = q \\ \mathrm{P}_{S_{\sim z}^{l}}\big(s(\mathsf{c})-i\big) & z \le i < q \\ 0 & \text{otherwise}. \end{cases} \tag{15}$$

For the message from the hidden VNs to the CNs in $\mathcal{CN}_z$, as it can be seen in (3), the lower can be evaluated in two steps. First U_1, the maximum among the L-bound from tests, is determined and then this is compared with $\mathsf{L}_{\mathsf{f}\to z}^{(\ell)}$ computed from the items. The pmf of U_1 is the $(d_{\mathsf{v},\mathsf{z}}-1)^{\text{th}}$ order statistics of $d_{\mathsf{v},\mathsf{z}}-1$ i.i.d discrete random variables with pmf $\mathrm{P}_{\mathsf{L}_{\mathsf{c}'\to z}^{(\ell-1)}|Z}\big(i|z\big)$. This is given by [15, page 42]

$$\begin{aligned}\mathrm{P}_{U_1|Z}\big(u_1|z\big) =& \mathrm{F}_{\mathsf{L}_{\mathsf{c}'\to z}^{(\ell-1)}|Z}\big(u_1|z\big)^{d_{\mathsf{v},\mathsf{z}}-1} \\ &- \mathrm{F}_{\mathsf{L}_{\mathsf{c}'\to z}^{(\ell-1)}|Z}\big(u_1-1|z\big)^{d_{\mathsf{v},\mathsf{z}}-1}.\end{aligned}$$

In the second step, we have to compute the pmf of the maximum of two independent random variables with pmfs $\mathrm{P}_{U_1|Z}(u_1|z)$ and $\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to z}|Z}(\mu|z)$. This is given by

$$\begin{aligned}\mathrm{P}_{\mathsf{L}^{(\ell)}_{z\to\mathsf{c}}|Z}(i|Z) =&\mathrm{P}_{U_1|Z}(i|z)\mathrm{F}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to z}|Z}(i|z)+\\ &\mathrm{F}_{U_1|Z}(i-1|z)\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to z}|Z}(i|z)\,,\end{aligned} \tag{16}$$

since the maximum is i if $\{U_1 = i \text{ and } \mathsf{L}^{(\ell)}_{\mathsf{f}\to z} \le i\}$ or $\{U_1 < i \text{ and } \mathsf{L}^{(\ell)}_{\mathsf{f}\to z} = i\}$.

We can apply similar reasoning for the U-bound, where we compute the minimum instead and have

$$\begin{aligned}\mathrm{P}_{\mathsf{L}_1|Z}(y|z) = \sum_{k=1}^{d_{\mathsf{v},\mathsf{z}}-1} &\mathsf{BN}\left(d_{\mathsf{v},\mathsf{z}}-1,k,\mathrm{F}_{\mathsf{U}^{(\ell)}_{\mathsf{c}\to z}|Z}(y|z)\right)-\\ &\mathsf{BN}\left(d_{\mathsf{v},\mathsf{z}}-1,k,\mathrm{F}_{\mathsf{U}^{(\ell)}_{\mathsf{c}\to z}|Z}(y-1|z)\right)\end{aligned} \tag{17}$$

and

$$\begin{aligned}\mathrm{P}_{\mathsf{U}^{(\ell)}_{z\to\mathsf{c}}|Z}(i|Z) =&\mathrm{P}_{\mathsf{L}_1|Z}(i|z)\left(1-\mathrm{F}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to z}|Z}(i-1|z)\right)+\\ &\left(1-\mathrm{F}_{\mathsf{L}_1|Z}(i|z)\right)\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to z}|Z}(i|z)\,,\end{aligned} \tag{18}$$

where L_1 is the minimum among the $d_{\mathsf{v},\mathsf{z}}-1$ U-bounds from tests. To find the distribution of $S^u_{\sim\boldsymbol{z}}$ and $S^l_{\sim z}$, we need to know the vector $\boldsymbol{z}_\mathsf{c}$ consisting of $d_{\mathsf{c},\mathsf{z}}$ bundles connected to CN c. We thus have two steps in evaluating $\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{c}\to z}|Z}(i|z)$ (and $\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{c}\to z}|Z}(i|z)$): First, generate $\boldsymbol{z}_\mathsf{c}$ and compute $\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{c}\to z}|Z,\boldsymbol{Z}_\mathsf{c}}(i|z,\boldsymbol{z}_\mathsf{c})$ for each distinct z in $\boldsymbol{z}_\mathsf{c}$, then evaluate the mean for all possible realization of $\boldsymbol{Z}_\mathsf{c}$, i.e,

$$\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{c}\to z}|Z}(i|z) = \sum_{\boldsymbol{z}_\mathsf{c}} \mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{c}\to z}|Z,\boldsymbol{Z}_\mathsf{c}}(i|z,\boldsymbol{z}_\mathsf{c})\mathrm{P}_{\boldsymbol{Z}_\mathsf{c}}(\boldsymbol{z}_\mathsf{c})\,. \tag{19}$$

Evaluating (19) is computationally infeasible. However, since the sum of $\boldsymbol{z}_\mathsf{c}$ equals the syndrome $s(\mathsf{c})$ which is distributed as $\mathsf{BN}(d_\mathsf{c},\gamma)$, it can be seen that for small values of γ the cdf of $S(\mathsf{c})$ approaches one very fast. For example, if we allow for an error $\varepsilon < 10^{-6}$ for $d_\mathsf{c} = 160$ and $\gamma = 1\%$ we have $1-\mathrm{F}_{S(\mathsf{c})}(s(\mathsf{c})) < \varepsilon$ for $s(\mathsf{c}) = 10$. The error is much smaller for lower values of γ and d_c. For such small values of syndrome we can easily list all possible realizations of $\boldsymbol{Z}_\mathsf{c}$. Noting that the order of permutation does not matter we can then compute the probability of the corresponding $(d_{\mathsf{c},\mathsf{z}},\boldsymbol{p})$ multinomial random variable, where $\boldsymbol{p}$ is a vector with $p_i \sim \mathsf{BN}(q,i,\gamma)$ for $i = 0\ldots q$.

5.2 Density Evolution for Item-Bundle Interaction

For the item-bundle interaction we have the density evolution as follows. As it can be seen in (7), a hidden CN f with $Z = z$, sends an L-bound of 1 to a VN x (which is defective) if the L-bound $\mathsf{L}^{(\ell)}_{z\to\mathsf{f}} = z$ and all the $q-z$ non-defective items have their U-bound $\mathsf{U}^{(\ell)}_{x\to\mathsf{f}} = 0$ (for the remaining $z-1$ defectives, $\mathsf{U}^{(\ell)}_{x\to\mathsf{f}} = 1$). Let N_0 denote the number of resolved items among the non-defective

members of a bundle. We thus have

$$\begin{aligned}\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x}|X,Z}\big(1|1,z\big) &= \mathrm{P}_{\mathrm{L}^{(\ell-1)}_{z\to\mathsf{f}}|Z,X}\big(z|z,1\big)\mathrm{P}_{N_0|Z,X}\big(q-z|z,1\big)\\ &= \mathrm{P}_{\mathrm{L}^{(\ell-1)}_{z\to\mathsf{f}}|Z}\big(z|z\big)\mathrm{P}_{N_0|Z}\big(q-z|z\big) \qquad (20)\\ &= \mathrm{P}_{\mathrm{L}^{(\ell-1)}_{z\to\mathsf{f}}|Z}\big(z|z\big)\left(\mathrm{P}_{\mathsf{U}^{(\ell-1)}_{x'\to\mathsf{f}}|X}\big(0|0\big)\right)^{q-z}. \qquad (21)\end{aligned}$$

The first equality comes from the fact that if Z is known, the number of resolved items among the non-defectives will not be affected if $X = 1$. The second equality is due to the fact that N_0 is distributed as $\mathsf{BN}\left(q-z, \mathrm{P}_{\mathsf{U}^{(\ell-1)}_{x'\to\mathsf{f}}|X}\big(0|0\big)\right)$. We can therefore compute $\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x},X}\big(1,1\big)$ as

$$\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x},X}\big(1,1\big) = \sum_{z=1}^{q}\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x}|X,Z}\big(1|1,z\big)\mathrm{P}_{X,Z}\big(1,z\big)\,,$$

with

$$\begin{aligned}\mathrm{P}_{X,Z}\big(1,z\big) &= \mathrm{P}_X\big(1\big)\mathrm{P}_{Z|X}\big(z,1\big)\\ &= \gamma\binom{q-1}{z-1}\gamma^{z-1}(1-\gamma)^{q-z}\,,\end{aligned}$$

giving $\mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x}|X}\big(1|1\big) = \mathrm{P}_{\mathsf{L}^{(\ell)}_{\mathsf{f}\to x},X}\big(1,1\big)/\mathrm{P}_X\big(1\big)$. Similarly, from (8), a CN f with $Z = z$, sends $\mathrm{U}^{(\ell)}_{\mathsf{f}\to x} = 0$ to a non-defective item if the U-bound $\mathsf{U}^{(\ell-1)}_{z\to\mathsf{f}} = z$ and all the j defective items have $\mathrm{L}^{(\ell)}_{x\to\mathsf{f}} = 1$. This gives

$$\begin{aligned}\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to x}|X,Z}\big(0|0,z\big) &= \mathrm{P}_{\mathsf{U}^{(\ell-1)}_{z\to\mathsf{f}}|Z}\big(z|z\big)\mathrm{P}_{N_1|Z}\big(z|z\big)\\ &= \mathrm{P}_{\mathsf{U}^{(\ell-1)}_{z\to\mathsf{f}}|Z}\big(z|z\big)\left(\mathrm{P}_{\mathrm{L}^{(\ell-1)}_{x'\to\mathsf{f}}|X}\big(1|1\big)\right)^{z}. \qquad (22)\end{aligned}$$

We can thus write

$$\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to x},X}\big(0,0\big) = \sum_{z=0}^{q-1}\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to x}|X,Z}\big(0|0,z\big)\mathrm{P}_{X,Z}\big(0,z\big)\,,$$

with

$$\mathrm{P}_{X,Z}\big(0,z\big) = (1-\gamma)\binom{q-1}{q-z}\gamma^{z}(1-\gamma)^{q-1-z}\,.$$

We can then compute $\mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to x}|X}\big(0|0\big) = \mathrm{P}_{\mathsf{U}^{(\ell)}_{\mathsf{f}\to x},X}\big(0,0\big)/\mathrm{P}_X\big(0\big)$.

For the message from a VN x to the hidden CN f, from (9) the L-bound of a defective item, $\mathrm{L}^{(\ell)}_{x\to\mathsf{f}}$ will be 1 if at least one of the CNs in $\mathcal{CN}_x$ sends a L-bound of 1. Thus we have

$$\mathrm{P}_{\mathsf{L}^{(\ell)}_{x\to\mathsf{f}}|X}\big(1|1\big) = 1-\left(1-\mathrm{P}_{\mathrm{L}^{(\ell-1)}_{\mathsf{c}\to x}|X}\big(1|1\big)\right)^{d_{\mathsf{v},\mathsf{x}}} \qquad (23)$$

Similar arguments gives

$$\mathrm{P}_{\mathsf{U}^{(\ell)}_{x\to\mathsf{f}}|X}\big(0|0\big) = 1-\left(1-\mathrm{P}_{\mathsf{U}^{(\ell-1)}_{\mathsf{c}\to x}|X}\big(0|0\big)\right)^{d_{\mathsf{v},\mathsf{x}}}. \qquad (24)$$

Furthermore from (5) the pmf of $\mathsf{L}^{(\ell)}_{\mathsf{f}\to z}$ is obtained as the convolution of q random variables with $\mathrm{P}_{\mathsf{L}^{(\ell)}_{x\to\mathsf{f}}|X}\big(y|x\big)$. The same is true for $\mathrm{U}^{(\ell)}_{\mathsf{f}\to z}$.

5.3 Density Evolution for Test-Item Interaction

For the test-item interaction we have the following probabilities. From (10), the L-bound from a CN in $\mathcal{CN}_x$ to a defective item is 0 if at least one item among the other $d_{\mathsf{c}} - 1$ items is non-defective and sends an U-bound of 1 (i.e., it is unresolved). Thus we have

$$\mathrm{P}_{\mathsf{L}_{\mathsf{c}\to x}^{(\ell)}|X}(1|1) = \left(1 - (1-\gamma)\mathrm{P}_{\mathrm{U}_{x'\to \mathsf{c}}^{(\ell-1)}|X}(0|0)\right)^{d_{\mathsf{c}}-1} . \tag{25}$$

Similarly, for the message to a non-defective item we have

$$\mathrm{P}_{\mathrm{U}_{\mathsf{c}\to x}^{(\ell)}|X}(0|0) = \left(1 - \gamma\mathrm{P}_{L_{x'\to \mathsf{c}}^{(\ell-1)}|X}(1|1)\right)^{d_{\mathsf{c}}-1} . \tag{26}$$

Referring to (12), the L-bound from a defective item to a CN c is 0 if the L-bound from a CN f is zero and none of the other CNs c sends an L-bound of 1. We then have

$$\begin{aligned}&\mathrm{P}_{\mathsf{L}_{x\to \mathsf{c}}^{(\ell)}|X}(1|1) = \\ &1 - \left(1 - \mathrm{P}_{L_{\mathsf{c}'\to x}^{(\ell-1)}|X}(1|1)\right)^{d_{\mathsf{v},\mathsf{x}}-1} \left(1 - \mathrm{P}_{\mathsf{L}_{\mathsf{f}\to x}^{(\ell-1)}|X}(1|1)\right) .\end{aligned} \tag{27}$$

With similar reasoning we have

$$\begin{aligned}&\mathrm{P}_{\mathrm{U}_{x\to \mathsf{c}}^{(\ell)}|X}(0|0) = \\ &1 - \left(1 - \mathrm{P}_{U_{\mathsf{c}'\to x}^{(\ell-1)}|X}(0|0)\right)^{d_{\mathsf{v},\mathsf{x}}-1} \left(1 - \mathrm{P}_{\mathsf{U}_{\mathsf{f}\to x}^{(\ell-1)}|X}(0|0)\right) .\end{aligned} \tag{28}$$

6 Results and Discussion

In this section, we present numerical results obtained from our asymptotic density evolution analysis and compare the thresholds with results from finite length simulations. In Table 1, we give the threshold γ_{th}, i.e., the maximum fraction of

Table 1: γ_{th} for $\Omega = 5\%$ for different values d_{v} and $d_{\mathsf{v},\mathsf{x}}$

q	$d_{\mathsf{v},\mathsf{x}}$	$d_{\mathsf{v}} = 4$	$d_{\mathsf{v}} = 5$	$d_{\mathsf{v}} = 6$	$d_{\mathsf{v}} = 7$	$d_{\mathsf{v}} = 8$
1		0.598	0.641	0.646	0.635	0.618
4	2	0.590	0.660	0.694	0.706	0.702
5	2	0.592	0.672	0.725	0.746	0.744
10	3	0.549	0.636	0.693	0.774	0.694

defective items that can be successfully resolved for $q = 4$, 5, and 10 for $\Omega = 5\%$ and different values of d_{v}. The results are obtained from the density evolution equations. The conventional setting $q = 1$ is also shown for comparison. For the three considered values of q, the best performance is achieved for $d_{\mathsf{v}} = 7$. The q with best performance, however, is not the same for all d_{v}. For each q and d_{v}, the value of $d_{\mathsf{v},\mathsf{x}}$ is optimized to yield the best performance. The best performance is achieved for $d_{\mathsf{v},\mathsf{x}} = 2$ for $q = 4, 5$ and for $d_{\mathsf{v},\mathsf{x}} = 3$ for $q = 10$.

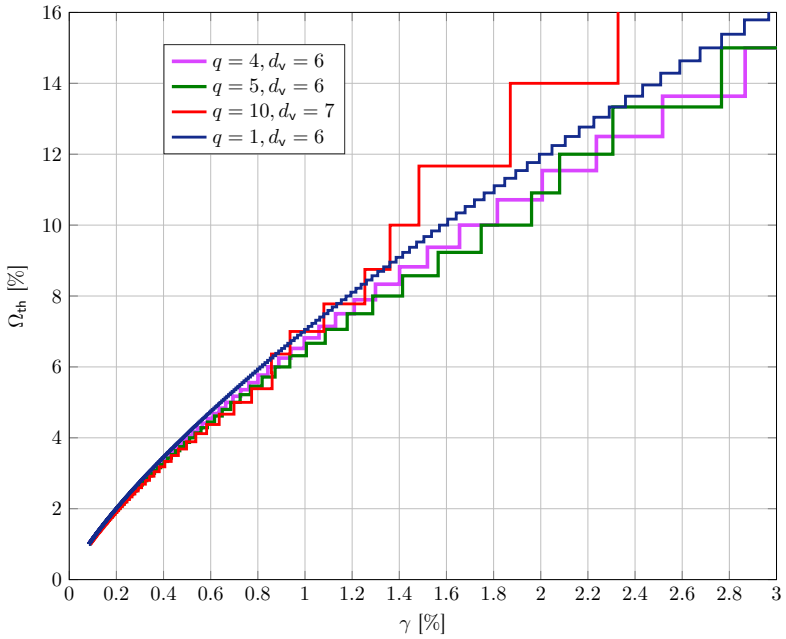


Figure 2: Minimum Ω for different values of γ for various bundle sizes q.

This means that adding more tests in $\mathcal{CN}_z$ helps in resolving bundles, but to get the resolution down to the level of items we need some tests to resolve at least some of the items. This explains why $d_{v,x} < 2$ does not work well.

In Fig. 2, we plot the threshold Ω_{th}, i.e., the minimum achievable rate obtained from the density evolution equations, as a function of γ . With $q = 10$ the performance is best for small values of γ, but poor for higher values. We conjecture that this is due to the suboptimality of BP decoding, and that spatial coupling will solve the problem. On the other hand, $q = 5$ performs better than the baseline system $q = 1$ for all γ.

In Fig. 3, we plot the misdetection rate of the proposed scheme with $q = 5$ and $q = 10$ as a function of γ and fixed rate $\Omega = 5\%$ (corresponding to $m = 10500$) for a finite length simulation with $n = 210000$. For comparison, we also plot the performance of the LDPC code-based scheme in [12], and the GLDPC code-based scheme in [4]. As predicted by the thresholds in Table 1, the proposed LDPC code-based GT scheme with non-binary variables performs significantly better than the LDPC code-based scheme in [12] (i.e., for $q = 1$). Furthermore, the LDPC code-based schemes perform significantly better than the GLDPC code-based scheme in [4].

Our results demonstrate that the grouping of items into bundles within the test matrix $\boldsymbol{A}$ of a quantitative GT scheme allows us to apply efficient non-binary message passing decoding with improved performance at an affordable complexity. The scheme is compatible with standard testing, and the structure of the test matrix is only slightly affected by the bundling while the overall weights of the columns or rows of $\boldsymbol{A}$ can be preserved.

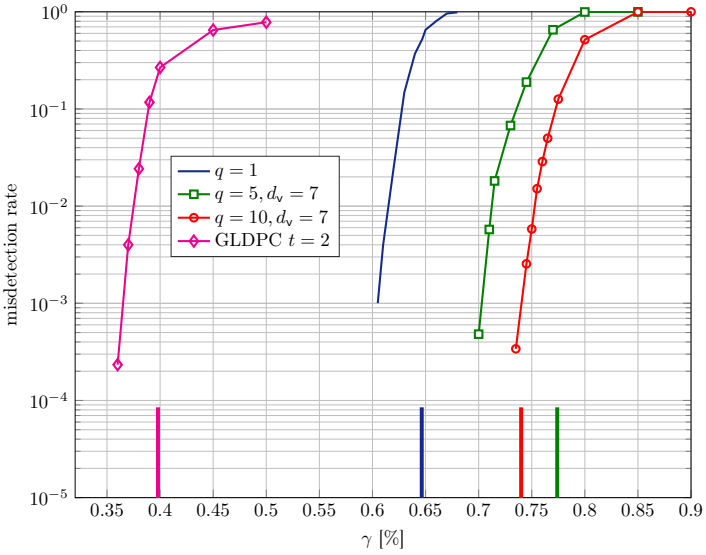


Figure 3: Simulation results showing the misdetection rate for various values of γ for a fixed rate $\Omega = 5\%$ for $n = 210000$. The GLDPC results are also shown for comparison. The vertical lines from the bottom mark the thresholds.

References

[1] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[2] M. T. Goodrich, M. J. Atallah, and R. Tamassia, "Indexing information for data forensics," in *Applied Cryptography and Network Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 206–221.

[3] A. Graell i Amat, G. Liva, E. Paolini, and C. Stefanovic, "Coded slotted aloha with stopping set resolution: A group testing approach," in *Proc. Asilomar Conf. Signals, Systems, and Computers (ACSSC)*, 2022.

[4] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *Proc. IEEE Inf. Theory Work. (ITW)*, 2019.

[5] ——, "Non-adaptive quantitative group testing using irregular sparse graph codes," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, 2019.

[6] E. Dutkiewicz, X. Costa-Perez, I. Z. Kovacs, and M. Mueck, "Massive machine-type communications," *IEEE Network*, vol. 31, no. 6, pp. 6–7, 2017.

[7] H. A. Inan, S. Ahn, P. Kairouz, and A. Ozgur, "A group testing approach to random access for short-packet communication," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019.

[8] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 185–191, 1985.

[9] A. G. Dyachkov and V. V. Rykov, "A coding model for a multiple-access adder channel." *Probl. Inf. Transm.*, vol. 17, no. 2, pp. 26–38, 1981.

[10] B. S. Tsybakov, "Resolution of a conflict of known multiplicity," *Probl. Inf. Transm.*, vol. 16, no. 2, pp. 69–82, 1980.

[11] G. del Angel. and T. L. Fine, "Solving conflicts of known multiplicity," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, 2002.

[12] M. M. Mashauri, A. Graell i Amat, and M. Lentmaier, "Low-density parity-check codes and spatial coupling for quantitative group testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2023.

[13] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: A novel counter architecture for per-flow measurement," in *Proc. Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS)*, 2008.

[14] E. Rosnes and A. Graell i Amat, "Asymptotic analysis and spatial coupling of counter braids," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7242–7263, 2018.

[15] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja, *A First Course in Order Statistics*. Society for Industrial and Applied Mathematics, 2008.