

Walter Nitzold

Rate-Flexible LDPC Convolutional Code Design



Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung

Nr. 78

**Walter Nitzold**

**Rate-Flexible  
LDPC Convolutional Code Design**

 VOGT

Dresden 2015

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im  
Internet über <http://dnb.dnb.de> abrufbar.

Bibliographic Information published by the Deutsche Nationalbibliothek  
The Deutsche Nationalbibliothek lists this publication in the Deutsche  
Nationalbibliografie; detailed bibliographic data are available on the  
Internet at <http://dnb.dnb.de>.

Zugl.: Dresden, Techn. Univ., Diss., 2015

Die vorliegende Arbeit stimmt mit dem Original der Dissertation  
„Rate-Flexible LDPC Convolutional Code Design“ von Walter Nitzold  
überein.

© Jörg Vogt Verlag 2015  
Alle Rechte vorbehalten. All rights reserved.

Gesetzt vom Autor

ISBN 978-3-938860-95-3

Jörg Vogt Verlag  
Niederwaldstr. 36  
01277 Dresden  
Germany

Phone: +49-(0)351-31403921  
Telefax: +49-(0)351-31403918  
e-mail: [info@vogtverlag.de](mailto:info@vogtverlag.de)  
Internet : [www.vogtverlag.de](http://www.vogtverlag.de)

Technische Universität Dresden

# **Rate-Flexible LDPC Convolutional Code Design**

**Walter Peter Nitzold**

von der Fakultät Elektrotechnik und Informationstechnik  
der Technischen Universität Dresden  
zur Erlangung des akademischen Grades

**Doktoringenieur**

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender: Prof. Dr.-Ing. Michael Schröter

Gutachter: Prof. Dr.-Ing. Dr. h.c. Gerhard Fettweis Tag der Einreichung: 02.03.2015  
Prof. Dr. Michael Lentmaier Tag der Verteidigung: 04.06.2015



To my father



# Abstract

Digital communication systems have advanced tremendously over the last decades and yielded applications such as high speed data transfer in LTE. In recent years, new communication scenarios like machine-to-machine systems have evolved. These scenarios demand for highly energy efficient receiver design as battery life is crucial in, e.g., sensor networks. A large part of the receiver's energy is consumed by the channel decoder, therefore code designs should be at hand, that exhibit low decoding complexity without any loss in transmission performance. As the aforementioned applications also demand for a flexible adaptation of the coding rate for efficient transmission, code designs have to be optimized for a wide rate interval. The recent ascent of LDPC convolutional codes as a capacity achieving code construction makes them a promising option for these communication systems. The thesis is concerned with the investigation of LDPC convolutional code constructions that on the one hand exhibit a good performance close to the Shannon limit for a wide range of rates as well as adequate decoding complexity.

Already regular LDPC convolutional codes are capable of achieving the Shannon limit when coupling length and width are chosen appropriately large. Their flexibility for achieving different rates is discussed and it is shown that with finite small coupling width, not the complete rate region can be covered smoothly with low complexity good performing code ensembles. Therefore, a new code construction is introduced that overcomes this issue by using slight irregularity in the ensemble description. These code ensembles outperform their regular counterparts for every considered rate.

The constraint of rate-compatibility adds a new restriction to the optimization of LDPC convolutional code constructions. To assess the different rate-compatible extension structures, a framework based on multi-edge type LDPC codes is introduced. The capabilities of regular rate-compatible LDPC convolutional codes are discussed and reveal that a break-down in performance for lower rates is always due to increasing variable node degree and can only be overcome by an increased coupling width. Based on this observation, new code constructions with relaxed degree evolutions for different rates are introduced and show significant performance increases and lower decoding complexity.

A similarity between the parity-check matrix structure of a nested rate-compatible code and an LDPC convolutional code is investigated. It turns out that the goal of transferring the good message propagation effects responsible for the performance improvement of LDPC convolutional codes to the nested codes can not be accomplished completely. A new double-banded parity-check matrix code construction is introduced with good decoding performance but capacity approaching effect is out of reach due to the lack of self-similarity in the decoding graph.



# Acknowledgements

During my time at the Vodafone chair for Mobile Communication Systems I learned many and various things and got to know different people from all over the world. Therefore, I have to be very grateful to many but can unfortunately only name few.

Gerhard Fettweis trusted me to become a part of his team and start the adventure that taught me so much. Thank you for all the possibilities that opened and will open up for me because of this.

I cannot express more gratitude to a supervisor than to Michael Lentmaier. From the beginning of our collaboration, he was a mentor in many ways. His attitude towards science deeply influenced me and I will always remember our various and long discussions not only at work and not only about work.

The work and time at the chair was embellished with the presence of many colleagues. Just to name a few for saying thank you: Nicola Michailow, Rohit Datta, Stefan Krone, Jan Dohl, Eckhard Ohlmer, Vincent Kotzsch, Peter Rost, Björn Almeroth, Ines Riedel, Alexandros Pollakis, Andreas Festag, Meik Dörpinghaus, Najeeb ul Hassan, Richard Fritzsche, Luciano Mendes, Steffen Watzek, Ivan Gaspar, Lukas Landau, Maximilian Matthe, Esther Perez Adeva.

I like to thank my family for their unconditional support. Thanks also go to my mother Karla Nitzold and my siblings Hagen, Liane, Gunnar, Christine and Nora for their interest and encouragement in this project. I explicitly thank my father Peter Nitzold for implanting the urge in me to understand things. He set the seed that has grown into this work. That is why I am dedicating this thesis to him.

Last but not least, I thank my two little children Martha and Rufus for getting my head away from thoughts about work and into their world. I am grateful that they have shown me in such an unexpected way, that there is so much love. Finally, my wife deserves to get my most grateful thank you as she unconditionally supported me through these 5 years of work. Let's go ahead...

*I know a big secret, but I know it only in music and only through the music  
can I express it. How do I get to it?*

- Arvo Pärt, 1978



# Contents

<b>Abstract</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xvi</b>
<b>Acronyms</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Fundamentals</b>	<b>5</b>
2.1 Objectives . . . . .	5
2.2 Channel Models . . . . .	6
2.2.1 Binary Erasure Channel . . . . .	6
2.2.2 Binary Input Additive White Gaussian Noise Channel . . . . .	7
2.3 Channel Capacity . . . . .	7
2.4 LDPC Codes . . . . .	8
2.5 Belief Propagation Decoder and its Decoding Complexity . . . . .	17
2.6 Density Evolution . . . . .	19
2.7 LDPC Convolutional Codes . . . . .	22
2.8 Incremental Redundancy . . . . .	26
2.9 Summary . . . . .	29
2.10 Related Literature . . . . .	29

<b>3</b>	<b>Nested Codes with Convolutional Structure</b>	<b>31</b>
3.1	Objectives . . . . .	31
3.2	Simple Protograph Construction . . . . .	32
3.3	Time-Varying LDPC Convolutional Codes with Rate-Compatibility . . . . .	33
3.4	Nested Codes with Double-Banded Structure . . . . .	37
3.5	Summary and Conclusions . . . . .	41
3.6	Protograph Matrix for Time-Variant Rate-Compatible LDPC Convolutional Code . . . . .	41
<b>4</b>	<b>Regular and Nearly-Regular LDPC Convolutional Codes</b>	<b>43</b>
4.1	Objectives and Literature . . . . .	43
4.2	Rate-flexible Code Design with Regular LDPC Convolutional Codes . . . . .	44
4.2.1	Construction . . . . .	44
4.2.2	Complexity . . . . .	46
4.2.3	Density Evolution . . . . .	47
4.3	Nearly-Regular LDPC Convolutional Codes . . . . .	51
4.3.1	Construction . . . . .	51
4.3.2	Density Evolution and Complexity . . . . .	53
4.3.3	Simulation . . . . .	59
4.4	Summary and Conclusions . . . . .	61
<b>5</b>	<b>Rate-Compatible LDPC Convolutional Code Design</b>	<b>63</b>
5.1	Objectives and Literature . . . . .	63
5.2	Rate-Compatible Protograph Based Ensembles with Raptor-Like Graph Structure . . . . .	64
5.3	Rate-Compatible Multi-Edge Type Ensembles . . . . .	72
5.3.1	Regular Rate-Compatible Ensembles . . . . .	75
5.3.2	Rate-Compatible Ensembles with Relaxed Degree Profile . . . . .	78
5.3.3	Ensembles with Raptor-Like Back-Connections . . . . .	81
5.3.4	Ensembles with Limited Back-Connection Depth . . . . .	86
5.3.5	Extension to Rate-Compatible Ensembles with Arbitrary Rate . . . . .	91
5.4	Summary and Conclusions . . . . .	96

---

<b>6</b>	<b>Conclusions and Further Work</b>	<b>99</b>
6.1	Summary and Conclusions . . . . .	99
6.2	Further Work . . . . .	102
<b>A</b>	<b>Tables of Degree Sequences</b>	<b>105</b>
<b>B</b>	<b>Proofs for Irregular LDPC Convolutional Code Ensembles</b>	<b>109</b>
	<b>List of Figures</b>	<b>112</b>
	<b>List of Tables</b>	<b>116</b>
	<b>Bibliography</b>	<b>117</b>
	<b>Curriculum Vitae</b>	<b>124</b>



# Nomenclature

This section shall summarize the notational conventions used throughout the thesis. We denote  $\mathbb{N}$  as the set of natural numbers without zero, so  $\mathbb{N} = \{1, 2, \dots\}$ . Sets are generally denoted by calligraphic letters such as  $\mathcal{E}$ . A specific code is also denoted by calligraphic letters as it is defined as a set of codewords. If a code itself is meant or a set of codes shall be clear from the context. Non-scalar variables are denoted by underlined letters, e.g.,  $\underline{H}$ . Dimensions should be clear from the context. Further variables used in the thesis are given in the following.

$\mathbb{Z}$	Integers
$w$	Coupling width
$L$	Coupling length (termination length)
$\mathcal{S}$	Set of code ensembles
$\mathcal{S}_C$	Set of code ensembles according to constraint $C$
$E\{a\}$	Expectation
$N_p$	Number of variable nodes in a protograph
$M_p$	Number of check nodes in a protograph
$Q$	Lifting factor for protographs
$\mathcal{C}$	Code
$n$	Number of codebits (Number of variable nodes in a Tanner graph)
$k$	Number of information bits
$m$	Number of constraint equations in parity check matrix (Number of check nodes in the tanner graph)
$n_t$	Number of code bits at time instant $t$
$m_t$	constraint equations in parity check matrix (Number of check nodes in the tanner graph) at time instant $t$
$I$	Number of iterations
$i, j$	Index variables
$l$	iteration index
$m_e$	Number of edge types
$t$	Index for time instant
$L_I$	inner periodicity
$L_O$	outer periodicity
$a$	incremental redundancy step

---

$\alpha$	maximum number of incremental redundancy steps
$b$	width of the matrix band
$\underline{D}$	vector of information bits
$\underline{X}$	vector of coded bits (codeword)
$\underline{Y}$	vector of received bits
$\hat{X}$	decoded codeword
$\underline{0}$	all zero matrix/vector (dimensions according to context)
$J$	Variable node degree
$K$	Check node degree
$X$	channel input random variable
$Y$	channel output random variable
$p(X)$	probability distribution of $X$
$C$	channel capacity
$e^{\text{Sh}}$	Shannon limit for BEC
$E_b/N_0^{\text{Sh}}$	Shannon limit for BIAWGN channel
$\mathcal{L}$	log-likelihood ratio
$\mathcal{L}$	log-likelihood function
$R$	rate
$P_b$	bit error probability
$s$	number of sockets

# Acronyms

**AWGN** additive white Gaussian noise

**BEC** binary erasure channel

**BIAWGN** binary-input additive white Gaussian noise

**BMS** binary memoryless symmetric

**BP** belief propagation

**DE** density evolution

**HARQ** hybrid automatic repeat request

**HSPA** high speed packet access

**IR** incremental redundancy

**LDGM** low-density generator-matrix

**LDPC** low-density parity-check

**LDPCC** low-density parity-check convolutional

**LLR** log-likelihood ratio

**LT** Luby transform

**LTE** Long Term Evolution

**M2M** machine-to-machine

**MAP** maximum a-posteriori

**MET** multi-edge type

**ML** maximum likelihood

**NACK** not-acknowledged

**PEG** progressive edge-growth

**SC** spatially-coupled

**SNR** signal-to-noise ratio

# Chapter 1

## Introduction

The 21<sup>st</sup> century can without any doubt be entitled as the century of digital information for everyone. Services to provide information have become ubiquitous in all areas of our daily life such as mobile web browsing, getting information for the public transport or even the last-minute buy of a ticket for a musical concert. The revolutionary paradigm change from analog to digital communication was introduced in the middle of the last century when Claude E. Shannon formulated a mathematical theory [Sha48] around the idea of a unified information quantum which was referred to as *bit* and messages, consisting of bits that represent information. The notion of the communication model he introduced was concentrated around the *communication channel*, a medium that acts as a conveyor for information. Unfortunately, this medium induces errors onto the transmitted bits. To be able to reconstruct the original message, a mechanism was needed which is referred to as *channel coding*. Shannon showed, that codes exist which can achieve a fundamental limit of information transmission, called the *channel capacity* but unfortunately gave no way how these coding schemes should be constructed in practice. Since then, coding theorists aimed for this final frontier of performance until a vast breakthrough with the invention of Turbo Codes [BGT93] in 1993. These codes showed a practical construction with an implementable decoder that gets performance very close to the Shannon limit. Now the Shannon limit could be approached with a coding scheme of reasonable decoding complexity. Only shortly later, Gallagers low-density parity-check (LDPC) codes [Gal63] were rediscovered by MacKay [MN95] with similar performance. The final frontier for coding theorist did fall with the invention of *LDPC convolutional codes* by Jimenez and Zigangirov [JFZ99] where an analytical proof was given in [KRU11] for their capacity achieving characteristics. The advent of LDPC and Turbo codes has pushed performance of current communication systems and even lead to new applications. The speeds of, e.g. 300Mbps in Long Term Evolution (LTE) or 42 Mbits in high speed packet access (HSPA) nowadays are no longer out of reach from a practical perspective. Besides such communication systems, mainly focused on speed, new applications emerged on the market such as *machine-to-machine (M2M)* communication. These systems are not characterized by the speed of information delivery but other metrics such as latency, resilience and en-

ergy efficiency. Sensor nodes, e.g., have to remain in functional condition for several years without changing the battery. Some may now say that research on coding theory is dead since the arrival at Shannon's capacity limit but the opposite is true. Still many practical considerations are unanswered and this thesis shall shed light on one of them. As the part of channel coding has a major impact on the complexity of a receiver in, e.g., a sensor node its impact on energy efficiency can not be neglected. The optimization for energy efficient channel coding can be done in a twofold way. Either the optimization of a decoding algorithm can be pursued or the code constructions itself have to be optimized for good performance with low-complexity. The latter part is the focus of this thesis.

The principle of channel coding is to encode a message consisting of  $k$  information bits into a vector of  $n$  encoded bits by adding  $m = n - k$  additional redundant bits. The basic parameter of a channel code is the *code rate* denoted by  $R = k/n$ . This quantity characterizes the amount of redundant bits that have been added to the original information message size. Intuitively the construction of a channel code for the most energy efficient transmission can be achieved when the amount of redundant bits (*redundancy*) is always kept as low as possible to obtain the shortest possible transmission time while still a successful decoding is ensured with high probability.

Now, a transmission scheme can be imagined which first sends out as few bits as possible and if the receiver is not able to decode, subsequent redundancy will be transmitted additionally in an incremental fashion. With such a transmission scheme, the optimal amount of redundancy is always guaranteed. To use such a transmission scheme, the channel code has to be designed in a way to support this subsequent transmission. As every step of incremental redundancy (IR) can be referred to as an own channel code, we are concerned with the construction of code families and not only single codes. In the construction of these codes two metrics are of importance, performance and decoding complexity. The performance of the code construction shall be optimized to achieve or at least approach the theoretical limits. On the other hand, good performance often comes at the price of high decoding complexity so the metric of complexity should be kept low. The interplay of these two metrics is the integral part of this thesis.

To motivate the use of this code constructions, shortly two communication scenarios shall be introduced.

**HARQ in LTE** The ever growing increase in transmission speed of mobile communication standards has gained the need for efficient channel coding schemes that maximize the transmission rate. The incorporation of *hybrid automatic repeat request (HARQ)* in LTE supports this feature by subsequent transmission of redundancy based on the need at the receiver. To use this transmission scheme, code designs are needed, that can produce subsequent IR steps. Typically, a connection between transmitter and receiver is established and a message with highest code rate (lowest amount of redundancy) is sent to the receiver. Depending on the channel quality, the receiver can either decode the message or

fails to decode because the channel is too bad for successful decoding. In case of decoding failure the receiver sends a not-acknowledged (NACK) message to the transmitter, signaling that it needs further redundancy for decoding. This procedure is continued until a successful decoding at the receiver is possible. This assures that every receiver gets the minimal amount of redundancy for the given channel quality and therefore maximizes its information rate.

**Multicast transmission for M2M** Assuming a message (e.g., a configuration update) shall be transmitted to a vast amount of different sensor nodes in a given area. Every sensor node has to receive the same message but undergoes different channel qualities. A classical code design for this scenario would have to construct the code with a code rate as low as the receiver with the worst channel quality would need to successfully decode the message. Inherently, this is of disadvantage for the sensor nodes with good channel quality as they receive redundancy, they normally would not need for decoding but due to the construction have to collect for complete reception. This approach certainly shows a bad energy efficiency for the sensor nodes with good channels. To overcome this issue, the transmitter of the multicast message uses a code construction capable of subsequently transmitting redundancy in an incremental way. Then every receiver could just collect as much redundancy as needed depending on its instantaneous channel quality. The amount of redundancy per sensor node is minimized and therefore energy efficiency for every sensor node in the communication system is maximized.

## Scope, Outline and Contributions

The main focus of this thesis is on code constructions that can support different rates and even further are able to be used in the two exemplary transmission scenarios described above. As low-density parity-check convolutional (LDPC) codes have superior performance qualities, the thesis concentrates on these code constructions and tries to gain insight in the possibilities and constraints of LDPC code constructions for different rates and additionally with the added constraint of rate-compatibility. The two core metrics that are used to assess the usability of LDPC codes at different rates in this thesis are summarized in the following two questions.

- ▷ How close can an LDPC code construction get to Shannon limit for different rates?
- ▷ How high is the decoding complexity to achieve this performance?

The interplay between these two metrics shall be discussed in the chapters of this thesis as follows.

- In Chapter 2, the basic concepts, tools and definitions are given. LDPC codes and their different definitions used throughout the thesis are explained and the performance assessment for these are given. Additionally, LDPCC codes are defined and their remarkable performance features are discussed. Finally, a definition for IR and rate-compatible code construction shall lay the basis and terminology for later discussions.
- The content of Chapter 3 investigates the observation of a similarity between the banded structure of an LDPCC code and the structure of rate-compatible nested codes. The chapter starts with an illustrative example of the similarity and its initial performance assessment. These models are then generalized to understand the mechanisms that lead to the performance differences for both ensembles. Finally, the chapter concludes with a proposal for a code construction that helps to partly overcome the issues of the nested codes.
- The focus of Chapter 4 is drawn to the flexibility of achieving different rates with a subclass of LDPCC codes which are referred to as regular LDPCC codes. This specific type has the unique property that its structure is very homogeneous and additionally it can be proven that these codes achieve the Shannon limit. The key question is if one has to deviate from these code constructions or if every rate in the interval  $R \in [0, 1]$  can be achieved with reasonable complexity and performance. The second part of the chapter introduces a slight irregularity to achieve the aforementioned goal and a detailed assessment of the tradeoff between performance and complexity is given. The results within this chapter are partly based on [NFL14].
- Chapter 5 is adding an additional constraint to the construction of LDPCC codes for different rates in the form of rate-compatibility. We introduce a generic model for rate-compatible LDPC codes based on multi-edge type ensembles. Different options for connectivity as well as different degrees for subsequent redundancy steps are discussed and their constructive issues as well as the trade-off in performance and decoding complexity are discussed for the case of LDPC block as well as LDPCC codes. The chapter ends with an additional introduction of a more generalized ensemble that is capable of fine granularly set the desired rates while ensuring good performance and low decoding complexity. The results within this chapter are partly based on [NLF12].
- Chapter 6 summarizes the key findings of this thesis and gives an outlook on further research topics.

Each chapter focuses on a specific topic. Contributions and related literature are covered on the beginning of each individual chapter.

# Chapter 2

## Fundamentals

### 2.1 Objectives

The fundamental analysis of channel code constructions relies on certain assumptions and methods that will be shortly introduced within this chapter. The chapter is organized as follows.

- Section 2.2 is related to the fundamental communication scenario that all investigations are based on. As the channel model is crucial for the performance of the code designs, specific emphasis is put onto the characterization of two channel models used throughout the thesis.
- The fundamental performance limit used to compare different code constructions is the channel capacity and will be introduced in Section 2.3.
- The core of the work within this thesis is related to the analysis of LDPC codes. Therefore, Section 2.4 gives introduction and definition of different ensembles that will be used throughout the thesis.
- Besides the performance of a code, its complexity and especially the decoding complexity is of utmost importance. LDPC codes are typically decoded with belief propagation (BP) decoders and therefore a discussion on the complexity metric of this decoder that is used throughout the thesis is given in Section 2.5.
- To assess the performance of different code constructions a method called density evolution (DE) is used within the thesis and shortly introduced in Section 2.6 for the considered ensembles.
- The main concern of this thesis is the adaptation of code constructions for the use with LDPC codes. The principle of these codes and the remarkable performance improvement is introduced in Section 2.7

- The main focus of this thesis is the behavior of rate-flexibility and rate-compatibility within the constructions of LDPC codes. The basic notion of rate-compatibility and IR is introduced in Section 2.8.

Finally, the key facts of this chapter are summarized in Section 2.9. As terms and definitions are given during the course of this chapter, the related literature is introduced in the respective sections and given in a consolidated manner at the end of the chapter.

## 2.2 Channel Models

We assume a source that produces a vector  $\underline{D}$  of information bits  $D \in \{0, 1\}$  with length  $k$  and a channel encoder which generates a codeword  $\underline{X}$  of length  $n$  denoted by  $\underline{X} = \{X_1, \dots, X_t, \dots, X_n\}$ . The codeword is transmitted over the channel and received in a perturbed version as  $\underline{Y} = \{Y_1, \dots, Y_t, \dots, Y_n\}$ , which is then decoded by the channel decoder to yield an estimate  $\hat{\underline{X}}$  of the sent codeword. The simplistic communication model for this channel is shown in Figure 2.1. The investigations in this thesis focus on *binary*

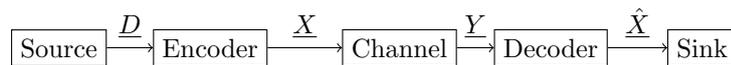


Figure 2.1: Simple communication model as derived in [Sha48]

*memoryless symmetric (BMS)* channels. In particular, we focus on the *binary erasure channel (BEC)* as well as the *binary-input additive white Gaussian noise (BIAWGN)* channel. The input to the channel is denoted by the random variable  $X$  which can take values from the alphabet  $\mathcal{X} = \{\pm 1\}$ . The output of a BMS is described by the random variable  $Y$  with  $Y \in \mathcal{Y}$ . The alphabet  $\mathcal{Y}$  can either be discrete or continuous as well as finite or infinite. Channel input and output at time  $t$  are referred to as  $X_t$  and  $Y_t$ , respectively. We consider only memoryless channels which are defined as follows.

**Definition 2.1** (Memoryless Channels [RU08]). *A channel, characterized by its transition probability  $p_{Y|X}(y|x)$  is said to be memoryless if*

$$p_{Y|X}(y|x) = \prod_t p_{Y_t|X_t}(y_t|x_t) \quad (2.1)$$

In the following, we shortly introduce the two channel models used throughout the thesis.

### 2.2.1 Binary Erasure Channel

The *binary erasure channel (BEC)* was introduced in 1954 as a toy example by Peter Elias [Eli54]. It is a remarkably simple but non-trivial model for a channel. The BEC

models a channel, where information bits are either received correctly or in error. While in the second case, information is completely lost, one can be sure to have received the bit correctly in the first case. Additionally, the occurrence of an error is known to the receiver in this channel model. The input alphabet for this channel is binary,  $\mathcal{X} = \{\pm 1\}$ , while the output alphabet is given by  $\mathcal{Y} = \{\pm 1, ?\}$ . Every transmitted bit is either received correctly with probability  $1 - \epsilon$  or erased with probability  $\epsilon$ . A BEC is fully characterized by the erasure probability  $\epsilon$  and can be visualized as depicted in Figure 2.2a.

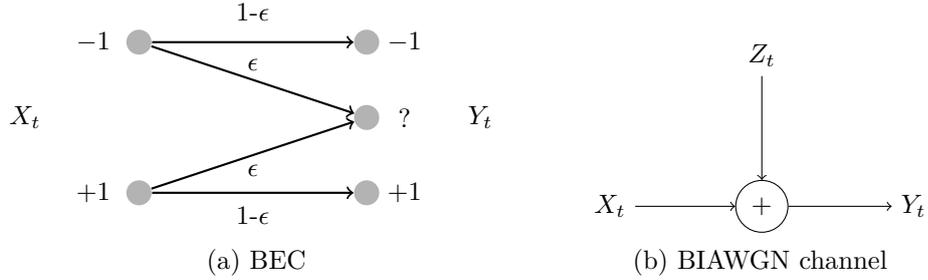


Figure 2.2: BMS channel models

### 2.2.2 Binary Input Additive White Gaussian Noise Channel

The *BIAWGN* channel uses a binary input alphabet  $\mathcal{X} = \{\pm 1\}$ . Its output is defined by the discrete time channel model

$$Y_t = X_t + Z_t \quad (2.2)$$

The additional noise term  $Z_t$  is additive white Gaussian noise (AWGN) with zero mean and variance  $\sigma^2$ , i.e.,  $Z_i \sim \mathcal{N}(0, \sigma^2)$ . Typically, the channel quality of the BIAWGN channel is parameterized with  $\sigma$ . A similar parameter is the energy per transmitted bit  $E_c$  to the noise energy  $\sigma^2$ . As we consider coded transmission throughout the thesis, we use the parameter of energy per information bit  $E_b = E_c/R$  that is normalized to the code rate. Finally, we get  $E_b/N_0 = E_c/(2R\sigma^2)$  with  $N_0 = 2\sigma^2$ .

## 2.3 Channel Capacity

Shannon's work asserts the existence of a fundamental transmission limit for every type of channel. We shortly reconsider its definition. The channel has input  $X$  with probability distribution  $p(X)$  and channel output  $Y$  with probability distribution  $p(Y)$ . Following the notation in [CT06] the mutual information of a channel with input  $X$ , output  $Y$  and transition probability  $p(Y|X)$  is defined as

$$I(X; Y) = H(Y) - H(Y|X) \quad (2.3)$$

where  $H(Y) = \mathbb{E}\{-\ln(P(Y))\}$  is the entropy of the channel output and  $H(Y|X) = \mathbb{E}\{-\ln(P(Y|X))\}$  is the conditional entropy of  $Y$  given  $X$ .

**Definition 2.2** (Channel Capacity). *The capacity of a channel with input  $X$ , output  $Y$  and transition probability  $p(y|x)$  is defined as*

$$C = \max_{\{p(X)\}} I(X; Y) \quad (2.4)$$

*which is the maximum mutual information, where the maximum is taken over the set of all input probability distributions  $p(X)$ .*

Shannon's channel coding theorem [Sha48] asserts, that transmission of information is possible at any rate  $R < C$  with vanishing bit error probability  $P_b$ . In the case of BEC, the capacity can be calculated relatively simple as

$$C_{\text{BEC}} = 1 - \epsilon \quad (2.5)$$

with  $\epsilon$  being the erasure probability of the channel. In case of the BIAWGN channel, the capacity is given by [Fri96]

$$C_{\text{AWGN}} = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-(z-v)^2/2} \log_2 \frac{2}{1 + e^{-2zv}} + e^{-(z+v)^2/2} \log_2 \frac{2}{1 + e^{2zv}} dz \quad (2.6)$$

with  $v = \sqrt{2RE_b/N_0}$ . The definition of Shannon's channel capacity gives rise to an asymptotic examination of the performance of a communication system. The highest rate, possible to communicate information is the channel capacity so if we set  $R = C$ , we can calculate *threshold* values  $\epsilon^{\text{Sh}}$  and  $E_b/N_0^{\text{Sh}}$  according to (2.5) and (2.6). These are the ultimate channel quality parameters where we can communicate at rate  $R$  and are called *Shannon limit*. The Shannon limit for the BEC and BIAWGN channel are depicted in Figure 2.3.

## 2.4 LDPC Codes

LDPC codes belong to the class of linear block codes and were invented by Gallager in 1963 [Gal63]. Since then they completely disappeared until their re-exploration in [MN95] when the Shannon-limit approaching performance could be shown and LDPC codes could compete with Turbo Codes. We will introduce regular LDPC codes first as a general example and elaborate on further more advanced code classes afterwards.

A linear block code can be defined by a parity-check matrix  $\underline{H}$ . Given a code of length  $n$ , a vector  $\underline{v} = (v_0, \dots, v_{n-1})$  is a codeword if and only if  $\underline{v}\underline{H}^T = \underline{0}$ .

**Definition 2.3** (LDPC and regular LDPC block codes [Len03]). *A binary block code of length  $n$ , defined by an  $m \times n$  parity check matrix  $\underline{H}$ , with  $m < n$  is an LDPC code if rows  $\underline{h}_i$  of  $\underline{H}^T$  are sparse, i.e.*

$$w_h(\underline{h}_i) \ll m, \quad i = 0, \dots, n-1 \quad (2.7)$$

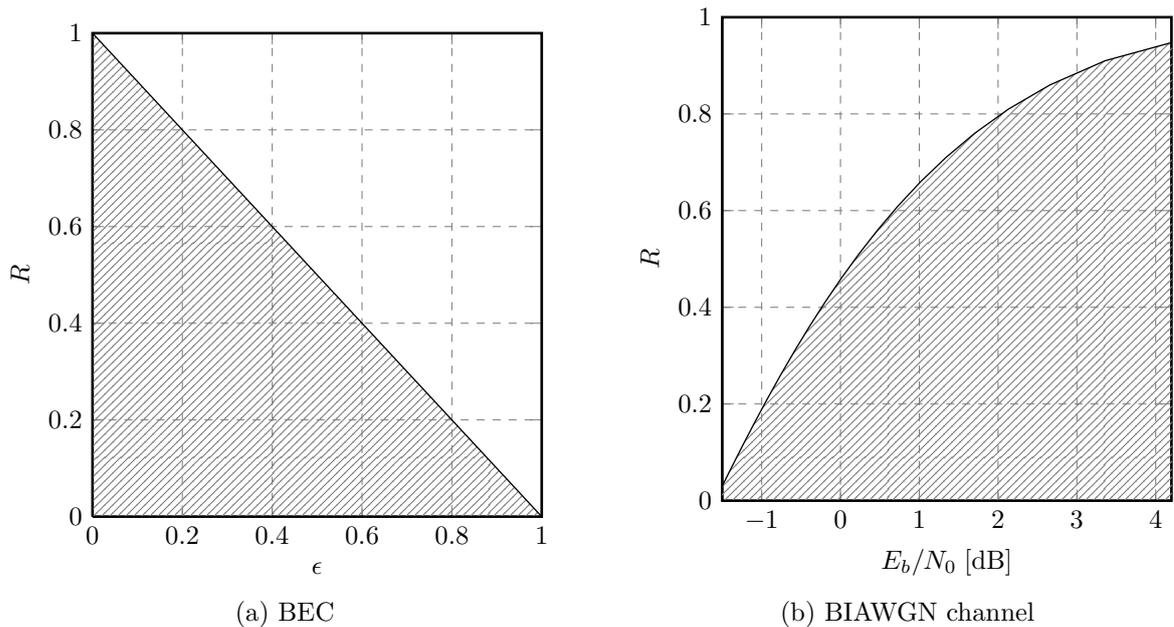


Figure 2.3: Shannon limit for BEC and BIAWGN channel. Shaded areas represent achievable regions for both cases.

where  $w_h(\cdot)$  denotes the Hamming weight [Fri96, Def. 1.7] of a given binary vector. Furthermore, if the parity-check matrix of an LDPC block code is restricted to have  $J$  ones in each column and  $K$  ones in each row of  $\underline{H}$ , the LDPC block code is called a regular  $(J, K)$  LDPC block code.

Given the above definition of a regular LDPC code, its design rate  $R$  is then defined as

$$R \geq 1 - \frac{J}{K}. \quad (2.8)$$

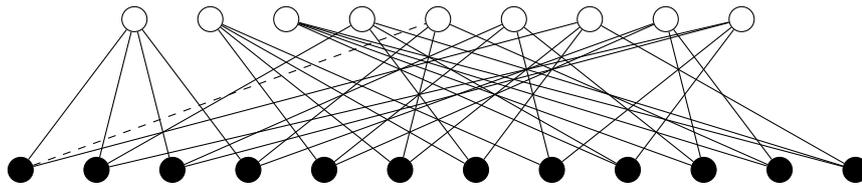
An LDPC code can be represented in different ways. While one is the parity-check matrix, the code can also be defined by a Tanner graph [Tan81]. A Tanner graph is a bipartite graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{C}, \mathcal{E}\}$  that consists of a set  $\mathcal{V}$  of variable nodes, a set  $\mathcal{C}$  of check nodes and a set  $\mathcal{E}$  of edges connecting variable and check nodes from  $\mathcal{V}$  and  $\mathcal{C}$ , respectively. An instance of a  $(3, 4)$ -regular LDPC code with length  $n = 12$  is depicted in both variants in Figure 2.4. Variable nodes (depicted by the black circles at the bottom of the Tanner graph) correspond to columns in the parity check matrix and check nodes correspond to rows. The edges in the Tanner graph are connected according to the positions of the one's in the parity check matrix. As an example, a one in column one and row five connects variable node one with check node five (as depicted with dashed lines in Figure 2.4).

## Ensembles and Finite Length Performance

To analyze the behavior of LDPC codes, we use the notion of *ensembles* of LDPC codes. These ensembles can be analyzed more easily and give an insight into the general behavior

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ \boxed{1} & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

(a) Parity-check matrix



(b) Tanner graph

Figure 2.4: Typical representations of an LDPC code

of a class of LDPC codes rather than one specific instance. In [RU01b] and [LMSS98] the notion of a *code ensemble* was introduced. We follow this approach as we are not interested in the performance of a particular code but the performance of an ensemble of codes where every code in the ensemble is characterized by the specific ensemble definition. A regular  $(J, K)$  LDPC code ensemble then consists of all parity-check matrices with a column weight of  $J$  and a row weight of  $K$ . Similarly this explanation of a code ensemble as the set of all codes that have the same characteristic according to the ensemble definition holds also for the ensemble definitions used throughout the thesis and introduced shortly. The great advantage of the investigations on code ensembles is that the average performance of codes in the ensemble can be calculated explicitly with the numerical method of DE. In [RU01b] we also find evidence, that if a code is drawn randomly from a code ensemble, its performance is close to the average performance of the ensemble. We stick to the performance assessment of code ensembles within this thesis and only occasionally show the performance of a particular code.

This finite length performance of a particular code with block length  $n$  can be described by two phenomena which can be observed when examining the bit error probability. On one hand, the waterfall region shows how close a code performs to Shannon limit. On the other hand, every code exhibits an error floor in the bit error rate curve. It is of great importance for a code designer to achieve a performance close to optimal Shannon limit as well as a very low error floor. It turns out, that getting good performance on both ends of the bit error rate plot is a hard task.

## Irregular ensembles

In contrast to regular LDPC code ensembles irregular LDPC code ensembles allow for more than one degree at variable and check node side. The notion of irregular LDPC ensembles was in depth investigated in [LMSS01a] [LMSS01b] [RU01b] [RSU01]. To accommodate for the different degrees on check and variable node side, irregular LDPC code ensembles are defined with degree distributions for check and variable node perspective. The degree distributions  $\alpha(x)$  for variable nodes and  $\gamma(x)$  for check nodes are defined as

$$\alpha(x) = \sum_{i \in \mathcal{J}} \alpha_i x^{J_i} \quad (2.9)$$

$$\gamma(x) = \sum_{i \in \mathcal{K}} \gamma_i x^{K_i} \quad (2.10)$$

where  $\mathcal{J}$  ( $\mathcal{K}$ ) denote the set of different variable (check) node degrees, and  $\alpha_i$  ( $\gamma_i$ ) refers to the fraction of variable (check) node of degree  $i$ . In the further analysis of irregular LDPC codes it is of advantage to also define degree distributions from an edge perspective.

$$\lambda(x) = \sum_{i \in \mathcal{J}} \lambda_i x^{J_i-1} \quad (2.11)$$

$$\rho(x) = \sum_{i \in \mathcal{K}} \rho_i x^{K_i-1} \quad (2.12)$$

where  $\lambda_i$  ( $\rho_i$ ) denote the fraction of *edges* connected to variable (check) nodes of degree  $i$ . While the node perspective is convenient when looking onto the construction of specific code ensembles, the edge perspective simplifies the performance evaluation of specific ensembles significantly. For detailed conversion rules between these two definitions, the reader may refer to [RU08].

**Example 2.1** (Irregular LDPC code ensemble). *Interpreting the Tanner graph in Figure 2.5a as an irregular LDPC ensemble, the degree distributions from a node perspective are given as*

$$\begin{aligned} \alpha(x) &= 1/3x^3 + 1/3x^2 + 1/3x \\ \gamma(x) &= x^3 \end{aligned}$$

The introduction of irregular degree distributions did allow the optimization of code ensembles for performance close to the Shannon limit. In [RSU01] and [CFRU01], the authors constructed irregular LDPC code ensembles that perform very close to Shannon limit for the BEC as well as the BIAWGN channel. To yield an irregular LDPC code ensemble for a given rate, the structure of the degree distributions was fixed. As the parameter space of the degree distributions form a convex polytope, a linear programming optimization problem can be formulated to find specific degree distributions with good properties. The structure of degree distributions was fixed in such a way, that the degree distributions on

the check node side were only allowed to have two adjacent degrees, i.e. check concentrated degree distributions. The variable node side on the other hand is only constrained to have a maximum degree. Interestingly with increasing maximum variable node degree the performance gets closer to Shannon limit but to approach this limit, a non-vanishing amount of degree-2 variable nodes is also needed. This gives rise to another problem which we briefly explain as follows.

As mentioned before, the performance of a code depends, amongst others, on the behavior in the error floor region. This behavior is strongly connected to the distance properties of a code ensemble. If the minimum distance [Fri96, Def. 1.8] grows linearly with the block length  $n$ , the code exhibits very low error floors. On the other hand, if distance grows only sub-linearly or logarithmically the error floor is expected to be high and will disturb the performance of an application designed for specific bit error rate regions. In [DRU06], the authors showed that for irregular LDPC code ensembles, the distance growth is directly connected to the degree distributions and especially to the amount of degree-2 variable nodes. Nevertheless, allowing for a specific amount of degree-2 variable nodes was still possible as long as the *stability condition* is satisfied. The stability condition is given in the following definition.

**Definition 2.4.** *Stability Condition [DRU06, Theorem 1] Given an irregular LDPC code ensemble with associated degree distribution pair  $(\lambda(x), \rho(x))$ , if the condition*

$$\lambda'(0)\rho'(1) < 1 \tag{2.13}$$

*is satisfied, then the minimum distance of the code ensemble grows linearly with probability at least  $1 - \ln \frac{1}{\sqrt{1 - \lambda'(0)\rho'(1)}}$ .*

Interestingly, a fixed fraction of degree two variable nodes can lead to very good performance due to a good threshold but one has to carefully design the degree distribution to not jeopardize the linear distance growth. Allowing an irregular LDPC ensemble to only have variable node degrees  $J \geq 3$  yields  $\lambda'(0)\rho'(1) = 0$  and therefore ensures linear distance growth with probability 1. The benefit of defining an irregular LDPC ensemble via its degree distribution is the simplification of the asymptotic analysis.

## Protograph Ensembles

To simplify the design and implementation of LDPC code ensembles while still retaining a tractable analysis, the emphasis went on to the construction of smaller matrices or graph prototypes that later can be expanded to a full parity-check matrix [RL09]. These code designs were first introduced in [Tho03] and are called *protographs*.

A protograph is a small graph  $\mathcal{C}_B$  that can be used to obtain a larger graph by a *copy-and-permute* procedure. The protograph is copied  $Q$  times to obtain  $Q$  replicas of each check

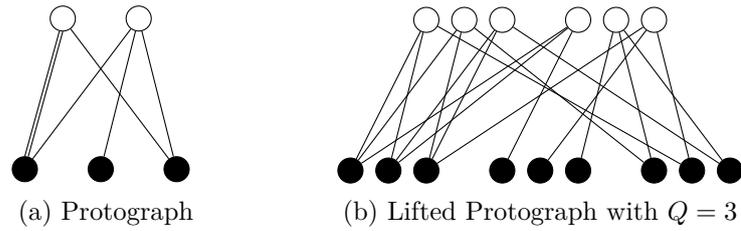


Figure 2.5: A protograph and one instance of a corresponding parity-check matrix with lifting factor  $Q = 3$ . Note that the permutations in the lifted protograph are assigned to each edge (double edges get a permutation per edge) and permutations can vary from edge to edge.

and variable node as well as  $Q$  replicas of each edge. Then edges are permuted between the different replicas of the nodes but only in a way that they still are connected to a replica of the node it was originally connected to in the protograph. This ensures that the degree profile of check and variable nodes remains the same as in the original protograph.

A protograph is defined as a bipartite graph  $\mathcal{C}_B = \{\mathcal{V}, \mathcal{C}, \mathcal{E}\}$  with a set  $\mathcal{V}$  of  $N_p$  variable nodes, a set  $\mathcal{C}$  of  $M_p$  variable nodes and a set  $\mathcal{E}$  of edges connecting the nodes from  $\mathcal{V}$  and  $\mathcal{C}$ . The rate of a protograph is defined as  $R = 1 - M_p/N_p$ . Also parallel edges between two nodes are allowed. Using the above described copy-and-permute procedure with  $Q$  copies, we can obtain a parity-check matrix of dimension  $m \times n$  with  $n = QN_p$  and  $m = QM_p$ . The edge permutations can also be modelled with a permutation matrix  $\Pi_{Q \times Q}$  of size  $Q \times Q$ . Note that the edge permutations can be done in different ways that might be beneficial for implementation and can be incorporated in the analysis and construction. Within the thesis, we consider random permutations of size  $Q \times Q$  although from an implementation perspective, circulants of size  $Q \times Q$  are preferable as they allow for easy parallelization of the decoding algorithm. The random permutations are used to justify the random ensemble definition for the analysis with DE. Similar to a Tanner graph, the protograph can also be represented by a matrix, the so called *base matrix*  $\underline{B}$ . Elements of the base matrix are assigned similar to the Tanner graph but with the addition that parallel edges in the protograph lead to integer elements greater than one.

**Example 2.2** (Protograph of rate  $R = 1/3$ ). *The protograph depicted in Figure 2.5 consists of  $N_p = 3$  variable nodes and  $M_p = 2$  check nodes and has therefore a design rate of  $R = 1/3$ . The figure also shows how permutations are assigned to the different edges. The base matrix is given as*

$$\underline{B} = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.14)$$

## Multi-edge Type Ensembles

The protograph ensembles have some limitations, e.g. when considering different numbers of variable nodes after lifting per variable node in the protograph. Such unequal lifting factors cannot be treated with the concept of protographs so a more generalized ensemble description was introduced in [RU02] and also discussed in [RU08]. While in the irregular ensemble definition the connectivity is only constrained by the node degrees, the *multi-edge type (MET) ensembles* define several edge classes. Every node in the graph is then defined by the number of sockets with which it connects to a specific edge class. An edge does only connect sockets of the same class. We slightly alter the notation in [RU08] for the general definition of MET ensembles.

A MET ensemble consists of  $m_e$  different edge types. A *degree type* of a check node is a vector of integers of length  $m_e$ . The  $i$ -th entry of this vector represents the number of sockets that are connected to edge type  $i$ . The degree type of a variable node consists of two parts. A length  $m_e$  vector fulfills the same purpose as on the check degree side. Additionally, variable nodes are related to the respective channel on which the codeword is transmitted. Therefore, we define a *received distribution* as a length  $m_r + 1$  vector. We can now assign a BMS channel to each  $i, i = 1, \dots, m_r$ . The channel for  $i = 0$  is used for punctured bits and therefore the associated channel is a BEC. The representation of the graph structure is done via a multinomial representation. We assume  $\underline{d} = (d_1, \dots, d_{m_e})$  to be a MET degree and let  $\underline{x} = (x_1, \dots, x_{m_e})$  be a vector of variables. We use  $\underline{x}^{\underline{d}}$  to denote  $\prod_{i=1}^{m_e} x_i^{d_i}$ . Additionally, let  $\underline{b} = (b_0, \dots, b_{m_r})$  be a received degree and  $\underline{r} = (r_0, \dots, r_{m_r})$  the corresponding vector of received variables. Typically, for received degrees only one entry is set to 1 and the rest is set to 0. With these definitions the MET ensemble is defined by the two multinomials

$$\nu(\underline{r}, \underline{x}) = \sum \nu_{\underline{b}, \underline{d}} \underline{r}^{\underline{b}} \underline{x}^{\underline{d}} \quad (2.15)$$

$$\mu(\underline{x}) = \sum \mu_{\underline{d}} \underline{x}^{\underline{d}} \quad (2.16)$$

with  $\nu_{\underline{b}, \underline{d}}$  and  $\mu_{\underline{d}}$  being nonnegative reals. Assuming a block length  $n$ , the quantity  $n\nu_{\underline{b}, \underline{d}}$  represents the number of variable nodes of degree type  $\underline{b}, \underline{d}$ . Similarly,  $n\mu_{\underline{d}}$  is the number of check nodes of degree type  $\underline{d}$ . Additionally, we define

$$\nu_{r_i}(\underline{r}, \underline{x}) = \frac{d\nu(\underline{r}, \underline{x})}{dr_i}, \quad \nu_{x_i}(\underline{r}, \underline{x}) = \frac{d\nu(\underline{r}, \underline{x})}{dx_i}, \quad \mu_{x_i}(\underline{x}) = \frac{d\mu(\underline{x})}{dx_i}. \quad (2.17)$$

To ensure that socket numbers for each edge type on check and variable node side are equal, we constrain the socket count for each edge type with

$$\nu_{x_i}(\underline{1}, \underline{1}) = \mu_{x_i}(\underline{1}), \quad i = 1, \dots, m_e. \quad (2.18)$$

Additionally, the received sockets also have to be constrained but as we consider only transmission over one similar channel for all variable nodes, the constraint reduces to  $\nu_{r_i}(\underline{1}, \underline{1}) = 1$ . The design rate of a MET ensemble is defined as

$$R = \nu(\underline{1}, \underline{1}) - \mu(\underline{1}). \quad (2.19)$$

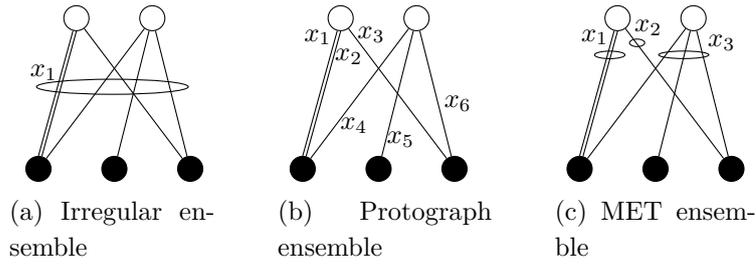


Figure 2.6: Interpretations of an exemplary graph structure with different ensemble descriptions.

Consider now an arbitrary enumeration of sockets on both variable and check node side with the total number of sockets  $s = s_1 + s_2 + \dots + s_{m_e}$ , where  $s_i$  is the number of sockets of edge type  $i$ . By connecting socket  $i$  to socket  $Pi(i)$  with a permutation  $\Pi$  on  $s$  letters we can define a particular graph. If we further restrict that  $i$  and  $\Pi(i)$  have to be of the same type the permutation can be decomposed into  $m_e$  permutations  $\Pi = (\Pi_1, \dots, \Pi_{m_e})$ . The MET ensemble is then defined by viewing  $\Pi_i$  as a random variable that is distributed uniformly over all permutations on  $s_i$  elements.

**Remark 2.1** (MET multinomials without received degree). *Within the thesis, we only consider the transmission of bits over the same channel and do not consider any punctured bits. As the multinomial representations especially in Chapter 5 are relatively complex, we simplify the notation. Therefore we omit the receive degree  $\underline{r}$  in  $\nu(\underline{r}, \underline{x})$  which simplifies the notation tremendously to  $\nu(\underline{x})$ . Note that for DE, the received degree  $r_1$  for the channel has to be multiplied to the multinomial for proper DE calculation as*

$$\nu(\underline{r}, \underline{x}) = r_1 \nu(\underline{x}). \quad (2.20)$$

**Example 2.3** (MET ensemble). *Taking the graph structure from Figure 2.5a and assign three different edge-types as depicted in Figure 2.6c, we get the multinomial expression as*

$$\nu(\underline{x}) = \nu_1 x_1^2 x_3 + \nu_2 x_3 + \nu_3 x_2 x_3 \quad (2.21)$$

$$\mu(\underline{x}) = \mu_1 x_1^2 x_3 + \mu_2 x_2^3 \quad (2.22)$$

**Example 2.4** (Protograph ensemble from Figure 2.5a as MET ensemble). *Taking the graph structure from Figure 2.5a and assume the graph to be a protograph with base matrix given by (2.14), we assign an edge-type per edge in the protograph, shown in Figure 2.6b. The associated MET multinomials are given as*

$$\nu(\underline{x}) = \nu_1 x_1 x_2 x_4 + \nu_2 x_5 + \nu_3 x_3 x_6 \quad (2.23)$$

$$\mu(\underline{x}) = \mu_1 x_1 x_2 x_3 + \mu_2 x_4 x_5 x_6 \quad (2.24)$$

**Example 2.5** (Irregular ensemble constructed from Figure 2.5a as MET ensemble). *Taking the graph structure from Figure 2.5a and assume the graph to be an irregular LDPC*

ensemble description, we only assign one edge-type to all edges as depicted in Figure 2.6a. We get the following multinomials

$$\nu(\underline{x}) = \nu_1 x_1^3 + \nu_2 x_1 + \nu_3 x_1^2 \quad (2.25)$$

$$\mu(\underline{x}) = \mu_1 x_1^3 + \mu_2 x_1^3 = (\mu_1 + \mu_2) x_1^3 \quad (2.26)$$

It turns out, that this ensemble is an irregular LDPC ensemble with check regular degree distribution.

To obtain the coefficients, for the above mentioned examples, one has to solve a linear equation system given by the socket constraints in (2.17) and the additional constraint  $\sum_i \nu_i = 1$ . Depending on the number of edge-types and coefficients, this equation system can be overdetermined, underdetermined or uniquely solvable. In the first case, a solution does not necessarily exist. The ensemble then does not yield a usable configuration. In the underdetermined case, an infinite number of solutions for the coefficients exist. This can be overcome by introducing additional constraints, e.g., to define a specific design rate for the given degree profile. If the system is uniquely solvable, only one configuration of degrees and coefficients is usable and the design rate of this ensemble is then a consequence of these degree profiles. Additional to the different cases mentioned above, negative solutions for coefficients also have to be avoided as these do not have any physical interpretation for the MET ensemble.

## Unstructured and Structured Ensembles

The ensemble definitions described above generally fall into two different categories. Regular  $(J, K)$  LDPC code ensembles and irregular  $(\lambda(x), \rho(x))$  LDPC code ensembles are *unstructured* ensemble definitions. Protograph and, to a certain extent, multi-edge type LDPC ensembles refer to *structured* ensembles.

The unstructured ensembles are characterized by a random construction of the parity-check matrices. A parity-check matrix drawn from this ensemble has no specific structure and only reassembles the degree distribution of the ensemble definition. The main disadvantage of these ensemble definitions is that a decoder implementation cannot benefit from simplifications of, e.g., storage of the parity-check matrix. On the other hand, these ensembles are important for the general analysis of LDPC ensembles as their simplistic definitions allow for a simple analysis of their performance.

The structured ensembles do overcome this disadvantage by imposing a desired structure onto the parity-check matrix. Protograph LDPC codes are a prominent example for such ensemble definitions. The incorporation of structure in the parity-check matrix does not neglect use of irregular degrees within the parity-check matrix and in fact, many standards use protograph ensembles that have a specific structure and assemble an overall irregular

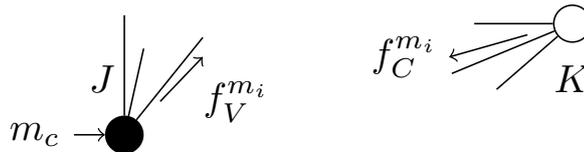


Figure 2.7: Visualization of the computations on variable (filled circle) and check (empty circle) node side for a BP decoding algorithm

parity-check matrix. The implementation of a decoder can then make use of the structure which results in a reduction of memory requirements and the support of other decoder architectures. Even parallel processing of decoder iterations is possible.

MET ensembles have a more general notion as they combine the idea of structured and unstructured ensembles. The introduction of different edge classes induces a given overall structure of the parity-check matrix but within one edge class, we still have the freedom of using irregular degree distributions.

## 2.5 Belief Propagation Decoder and its Decoding Complexity

While encoding of a codeword is usually very simple to implement [RU01a] the decoder is of utmost importance to be optimized for low implementation complexity. The typical decoder algorithm used for decoding of LDPC codes is the *BP decoder* [Gal63, Pea88]. Although its sub-optimality, this iterative decoding approach yields very good performance. In the following, we shall briefly explain the notion behind this message-passing algorithm and state the definition of decoding complexity that is used throughout the thesis for further evaluation. All simulations and discussions are based on the original sum-product algorithm by Gallager [Gal63]. Further simplifications for implementation are widely discussed but not directly connected to the construction of codes.

The decoding is based on the exchange of messages between the connected check and variable nodes in the Tanner graph. An output message on an edge is calculated on basis of the other incoming messages connected to the respective node the edge is emanating from. The principal behavior for check and variable nodes is depicted in Figure 2.7. The functionals for calculation can be separated in one type on the variable node side and one type on the check node side which always are functions of the messages on edges that are attached to the same node. We define two functionals to calculate a new message  $m_i$  for edge  $i$  as

$$m_i = f_V^{(m_i)}(m_1, \dots, m_{i-1}, m_{i+1}, \dots, J) \quad (2.27)$$

$$m_i = f_C^{(m_i)}(m_1, \dots, m_{i-1}, m_{i+1}, \dots, K) \quad (2.28)$$

where  $f_V^{(m_i)}$  denotes a variable node update and  $f_C^{(m_i)}$  denotes a check update.

**Example 2.6** (Gallagers sum product algorithm [Gal63]). *For the original sum-product algorithm introduced in [Gal63] the functionals for node updates and final decoding operation are given as*

$$f_V^{(m_i)} = m_c + \sum_{j \neq i} m_j \quad (2.29)$$

$$f_C^{(m_i)} = 2 \tanh^{-1} \left( \prod_{j \neq i} \tanh \left( \frac{1}{2} m_j \right) \right) \quad (2.30)$$

with  $m_c$  denoting the input message from the channel.

Using this abstract notion of a functional for decoding mechanisms within a message passing decoder, we assign a complexity metric to the computation of the variable node functional in the following definition

**Definition 2.5** (Unit of complexity). *The atomic unit for complexity is normalized to the computation of one functional on the variable node side and is defined as*

$$\mathfrak{C}(f_V^{(m_i)}) = 1 \quad (2.31)$$

where we assume that the computation of an outgoing message from a variable node has single unit computational complexity.

As this states the complexity of a single message computation, the complexity for a complete update of a variable node consisting of  $J$  edges (which in the Tanner graph denotes one bit of the codeword) within one iteration is given as

$$\mathfrak{C}(J f_V^{(m_i)}) = J. \quad (2.32)$$

Counting for the overall number of variable nodes  $n$  in the Tanner graph and the respective iterations  $I$  we get an overall decoding complexity as

$$\mathfrak{C} = n \cdot I \cdot \mathfrak{C}(J f_V^{(m_i)}) = n \cdot I \cdot J. \quad (2.33)$$

As in the remainder of the thesis, the investigations are of asymptotic nature assuming infinite iterations, we normalize the complexity to one iteration. Additionally, to decouple the complexity metric from the respective rate of a specific codeword, we normalize the given complexity metric to the number of information bits  $k$  and yield the final complexity expression as

$$\mathfrak{C} = \frac{n \cdot I \cdot J}{I \cdot k} = \frac{J}{R} \quad (2.34)$$

with the code rate  $R = k/n$ . With this, we decouple the implementation complexity from the specific decoding algorithm and therefore can later on draw conclusions on the computational complexity of LDPC ensembles based on their constructional properties only. The metric for complexity, derived and used within this thesis is motivated by [MCFF10].

## 2.6 Density Evolution

To analyze the iterative decoding performance of certain ensembles of codes such as irregular, protograph or MET ensembles under BP decoding, we shortly describe the method of *DE* which is the crucial analysis tool for the thesis. This method was first introduced in [LMSS98] and later refined for general channels in [RU01b]. For illustrative purposes we stick to the presentation in [RU08]. The messages to be exchanged in the BP algorithm are *log-likelihood ratios (LLRs)* which we define as follows. Given a BMS channel as in Definition 2.1 with transition probability  $p_{Y|X}(y|x)$ , the associated LLR function  $\mathfrak{L}(y)$  is given by

$$\mathfrak{L}(y) = \ln \frac{p_{Y|X}(y|1)}{p_{Y|X}(y|-1)} \quad (2.35)$$

which has an associated LLR  $\mathcal{L} = \mathfrak{L}(Y)$  for the random variable  $Y$ . As the LLR  $\mathcal{L}$  itself is a random variable, its probabilistic properties are fully defined by a density  $\mathbf{a}$  which we relate to as an L-density. Assuming now, we have two observations  $Y_1$  and  $Y_2$  resulting from transmission of  $X$  over two independent channels and associated LLRs  $\mathcal{L}_1$  and  $\mathcal{L}_2$ . Then it can be shown, that the LLR of the joint random variable  $(Y_1, Y_2)$  is given as  $\mathcal{L}_1 + \mathcal{L}_2$ . If  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are the associated L-densities then the L-density of  $(Y_1, Y_2)$  is simply the convolution of the individual densities as  $\mathbf{a}_1 \otimes \mathbf{a}_2$ . The operational meaning of this convolution is in fact the behavior how densities of LLRs are combined on variable nodes.

A similar observation can be made for the nodes on check node side. We first define the hard-decision function as

$$\mathfrak{H}(x) = \begin{cases} +1 & \text{if } x > 0 \\ +1 & \text{with probability } \frac{1}{2} \text{ if } x = 0 \\ -1 & \text{with probability } \frac{1}{2} \text{ if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (2.36)$$

With this definition we can define another quantity as

$$g(y) = (\mathfrak{H}(l(y)), \ln \coth(|\mathfrak{L}(y)|/2)) \quad (2.37)$$

with associated random variable  $G = g(Y)$ . The density of  $G$  is denoted by  $\mathbf{b}$ . As  $g(y)$  takes values in  $\{\pm 1\} \times [0, +\infty]$  the G-density  $\mathbf{b}(s, x)$  has the form

$$\mathbf{b}(s, x) = \mathbb{1}_{s=1} \mathbf{b}(1, x) + \mathbb{1}_{s=-1} \mathbf{b}(-1, x). \quad (2.38)$$

where  $\mathbb{1}_x$  is the indicator function. As G-densities are defined over the product of  $\mathbb{R}^+$  and  $\{\pm 1\}$  the convolution of these densities is two dimensional over the group  $\mathbb{F}_2 \times [0, +\infty]$ . We denote this convolution by the symbol  $\boxtimes$  and refer to it as the convolution in the check node domain. As mentioned before, the two major channel models considered in this thesis are the BEC and the BIAWGN channel. The associated L-densities for these two channels are given as follows.

**Definition 2.6** (L-density for BEC). *Given a BEC with erasure probability  $\epsilon$ , the input alphabet to be  $\{\pm 1\}$  and assuming that  $X = 1$  was sent, the L-density is given as*

$$\mathbf{a}_{BEC}(y) = \epsilon \Delta_0(y) + (1 - \epsilon) \Delta_{+\infty}(y) \quad (2.39)$$

with  $\Delta_0(y)$  and  $\Delta_{+\infty}(y)$  being point masses at zero and infinity respectively.

**Definition 2.7** (L-density for BIAWGN). *Given a BIAWGN channel with standard deviation  $\sigma$ , the corresponding L-density is given as*

$$\mathbf{a}_{AWGN}(y) = \sqrt{\frac{\sigma^2}{8\pi}} e^{-\frac{(y - \frac{\sigma^2}{8})^2 \sigma^2}{8}} \quad (2.40)$$

We now illustrate the process of DE with the example of regular LDPC codes. Consider first the actions on the variable node side as already illustrated in Figure 2.7. While the BP algorithm for decoding is concerned with the computation and exchange of LLRs, the asymptotic analysis of DE assumes an infinite block length and number of iterations which mathematically can be treated with the exchange of corresponding L-densities. We therefore now assume that edges emit not LLRs but their corresponding densities. In this respect we move from the analysis of a specific code to the analysis of an ensemble of codes. Assuming a  $(J, K)$  regular LDPC code, we have variable nodes with  $J$  edges attached. Let  $\mathbf{x}^{(l)}$  denote the L-density emitted from a variable node at iteration  $l$ . As the LLRs on variable nodes simply add up, the respective L-densities convolve. We therefore can state the outgoing L-density  $\mathbf{x}^{(l+1)}$  at iteration  $l + 1$  as

$$\mathbf{x}^{(l+1)} = \mathbf{c} \otimes \left( \mathbf{y}^{(l)} \right)^{\otimes (J-1)} \quad (2.41)$$

where the exponent stems from the fact that due to the regular degrees, we sum up equal LLRs resulting in a convolution of the same densities. Note that L-density  $\mathbf{c}$  corresponds to the L-density contributed by the channel. For the check nodes the LLRs are not simply summed up but combined according to (2.30). It turns out (details can be found in [RU08]), that the combination of LLRs at check node side results in a convolution of L-densities in the check node domain, denoted by the convolutional operator  $\boxtimes$ . With this observation, the update of densities  $\mathbf{y}^{(l)}$  at check node side in iteration  $l$  can be given as

$$\mathbf{y}^{(l)} = \left( \mathbf{x}^{(l)} \right)^{\boxtimes (K-1)} \quad (2.42)$$

Putting together (2.41) and (2.42), the definition of DE is given as follows.

**Definition 2.8** (DE for regular  $(J, K)$  LDPC code ensembles). *Given a BMS channel with L-density  $\mathbf{c}_{BMS}$ , the evolution of L-densities  $\mathbf{x}^{(l)}$  on the variable node side of a regular  $(J, K)$  LDPC code ensembles at iteration  $l$  is defined by*

$$\mathbf{x}^{(l+1)} = \mathbf{c}_{BMS} \otimes \left( \left( \mathbf{x}^{(l)} \right)^{\boxtimes (K-1)} \right)^{\otimes (J-1)} \quad (2.43)$$

with  $\mathbf{x}^{(0)} = \mathbf{c}_{BMS}$ .

In the same manner, the evolution of densities for an irregular LDPC code ensemble with degree distributions  $(\lambda(x), \rho(x))$  is given as follows.

**Definition 2.9** (DE for an irregular  $(\lambda(x), \rho(x))$  LDPC code ensemble). *Given a BMS channel with  $L$ -density  $\mathbf{c}_{BMS}$ , the evolution of  $L$ -densities  $\mathbf{x}^{(l)}$  on the variable node side of an irregular LDPC code ensemble with degree distributions  $(\lambda(x), \rho(x))$  at iteration  $l$  is defined by*

$$\mathbf{x}^{(l+1)} = \mathbf{c}_{BMS} \otimes \lambda\left(\rho\left(\mathbf{x}^{(l)}\right)\right) \quad (2.44)$$

where  $\lambda(\mathbf{x}) = \sum_{i=2}^{J_{max}} \lambda_i \mathbf{x}^{\otimes i-1}$  and  $\rho(\mathbf{x}) = \sum_{i=2}^{K_{max}} \rho_i \mathbf{x}^{\boxtimes i-1}$ .

Similarly, we can give the DE update equation for a generic MET ensemble defined by multinomials  $\nu(\underline{x})$  and  $\mu(\underline{x})$ . To state the recursive equation in a simple way, we first introduce the vector multinomial representation from an *edge perspective* as follows.

**Definition 2.10** (Vector multinomials from an edge perspective). *Given a MET ensemble with  $m_e$  edge-types and multinomials  $\nu(\underline{x})$  and  $\mu(\underline{x})$  from a node perspective, the associated vector multinomials from an edge perspective are given as*

$$\underline{\lambda}(\underline{x}) = \left( \frac{\nu_{x_1}(\underline{x})}{\nu_{x_1}(\underline{1})}, \frac{\nu_{x_2}(\underline{x})}{\nu_{x_2}(\underline{1})}, \dots, \frac{\nu_{x_{m_e}}(\underline{x})}{\nu_{x_{m_e}}(\underline{1})} \right) \quad (2.45)$$

$$\underline{\rho}(\underline{x}) = \left( \frac{\mu_{x_1}(\underline{x})}{\mu_{x_1}(\underline{1})}, \frac{\mu_{x_2}(\underline{x})}{\mu_{x_2}(\underline{1})}, \dots, \frac{\mu_{x_{m_e}}(\underline{x})}{\mu_{x_{m_e}}(\underline{1})} \right). \quad (2.46)$$

for variable and check nodes, respectively.

Utilizing this multinomial representation we define DE for MET ensembles as follows.

**Definition 2.11** (DE for MET LDPC ensembles). *Consider a MET ensemble with associated multinomials  $\underline{\lambda}(\underline{x})$  and  $\underline{\rho}(\underline{x})$ . Further, consider transmission over a BMS channel where  $\mathbf{c}_{BMS}$  denotes the  $L$ -density of received values. Note that, in simplification of the general case, we assume the same BMS channel for every edge-type. Let  $\underline{\mathbf{a}}^{(l)}$  denote the vector of densities (As every edge type conveys one particular density we consider now vectors of densities  $\underline{\mathbf{a}} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{m_e})$ ) passed from variable nodes to check nodes in iteration  $l$  and assuming that  $\underline{\mathbf{a}}^{(0)} = \mathbf{c}_{BMS}$ . Then for  $l \geq 1$ , the recursive equation for DE is given as*

$$\underline{\mathbf{a}}^{(l)} = \mathbf{c}_{BMS} \otimes \underline{\lambda}\left(\underline{\rho}(\underline{\mathbf{a}}^{(l-1)})\right). \quad (2.47)$$

Further, the density of the final decision of decoding at iteration  $l$  is given by  $\mathbf{a}_{BMS} \otimes \nu\left(\underline{\rho}(\underline{\mathbf{a}}^{(l)})\right)$ .

As protograph ensembles can be easily modeled with the MET framework, we leave out the derivation of the specific DE equation and refer the reader to [RU01b, Tho03]. While the computational burden for the general case which includes the BIAWGN channel can be lowered by methods described in Appendix B of [RU08], for the BEC the DE equations break down to very simple recursions and we exemplarily state the DE equation for the BEC and regular LDPC code ensembles in the following.

**Definition 2.12** (DE for regular LDPC codes on BEC). *We use the definitions of received  $L$ -densities for the BEC from Definition 2.6. Let  $x^{(l)}$  denote the probability of erasure at variable node side in iteration  $l$ . Then the recursive DE equation for an  $(J, K)$  regular LDPC code ensemble on the BEC with erasure probability  $\epsilon$  is defined by*

$$x^{(l)} = \epsilon \left( 1 - \left( 1 - x^{(l-1)} \right)^{K-1} \right)^{J-1}. \quad (2.48)$$

As we are concerned with the asymptotic behavior of the ensemble, we can use DE to determine channel parameters for which the remaining error probability approaches zero and other channel parameters for which it is strictly bounded away from zero. The goal of the analysis with DE is then to find the ultimate threshold  $\xi_{\text{BP}}$  (where "BP" denotes the assumption of BP decoding) such that for  $\xi > \xi_{\text{BP}}$ , reliable transmission is not possible as the remaining error probability is bounded away from zero and  $\xi \leq \xi_{\text{BP}}$  for which error probability always converges to zero. We formally state the definition of the so called iterative decoding threshold  $\xi_{\text{BP}}$  in the following.

**Definition 2.13** (Iterative decoding threshold). *Let  $Pr^{(l)}(\xi)$  denote the remaining error probability after  $l$  iterations given a channel model that is characterized by a channel parameter  $\xi$  which is ordered by degradation [RU08]. Then the iterative decoding threshold  $\xi^{BP}$  is given as*

$$\xi^{BP} = \sup\{\xi \in \Xi : Pr^{(l)}(\xi) \xrightarrow{l \rightarrow \infty} 0\} \quad (2.49)$$

where  $\Xi$  denotes the domain of  $\xi$ . For the BEC we have  $\xi = \epsilon$ ,  $\xi^{BP} = \epsilon^{BP}$  and  $\Xi = [0, 1]$ . For BIAWGN,  $\xi = \sigma$ ,  $\xi^{BP} = \sigma^{BP}$  and  $\Xi = \mathbb{R}^+$ . Equivalently,  $\sigma^{BP}$  can also be stated via  $E_b/N_0^{BP}$ .

## 2.7 LDPC Convolutional Codes

We shortly introduce the notion of *LDPCC codes* – also referred to as *spatially-coupled (SC) LDPC codes*<sup>1</sup> – as the remainder of the thesis will be concerned with performance tradeoffs between coupled and uncoupled versions of differently constructed code ensembles. For the sake of clarity of presentation, we point out the general idea with a coupled system of regular LDPC code ensembles with the help of the ensemble definition from [KRU11]. Nevertheless, the procedure of spatial coupling does easily extend to irregular, MET and protograph ensembles. These extensions will be noted briefly for completeness.

We assume a sequence of  $L$  time instants indexed by a time index  $t \in [0, L - 1]$  with  $L$  as the *coupling length*. At each time instant  $t$ , a  $(J, K)$  regular LDPC code ensemble with  $n_t$  variable nodes (code bits) and  $\frac{J}{K}n_t = m_t$  check nodes is located. For the remainder of the thesis, we assume  $n_t$  to be constant for  $t \in [0, L - 1]$  as well as  $m_t$  being constant for

<sup>1</sup> Note that the terms SC LDPC codes and LDPCC codes refer to the same code ensemble description but are equally used in literature. Within the thesis, we use both terms interchangeably.

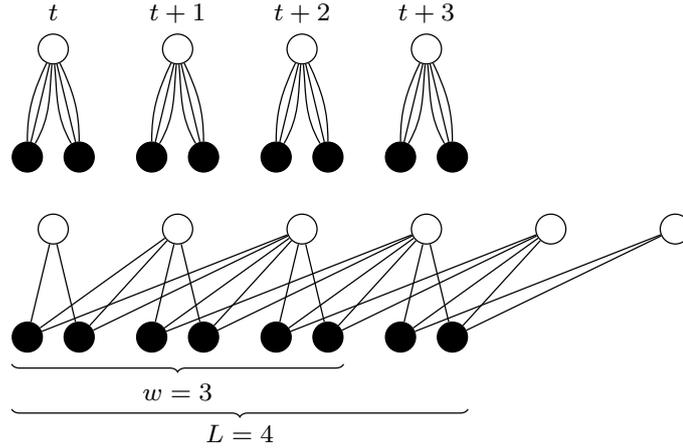


Figure 2.8: Spatially-coupled code ensemble

the complete sequence. Currently, the sequence of codes (or codewords) is a sequence of  $(J, K)$  block codes that do not interact with each other. As the idea of spatial coupling is to interconnect block codes over different time instants, it remains to define how interconnections between code ensembles at different time instants are chosen, i.e., how edges are distributed over time instants. We assume that every of the  $J$  edges of all variable nodes at time instant  $t$  is uniformly and independently connected to a check node in the range  $\{t, \dots, t + w - 1\}$ , where  $w$  denotes the *coupling width* and is a measure of the strength of coupling. Similar, every of the  $K$  edges of all check nodes at time instant  $t$  is connected to a variable node in the range  $\{t - w + 1, \dots, t\}$ .

A graphical visualization of a coupled chain of block codes with  $w = 3$  is shown in Figure 2.8. We refer to such an ensemble as a  $(J, K, L, w)$  LDPCC ensemble. Note that at position  $t \in \{L, \dots, L + w - 2\}$  we have to append additional check nodes as depicted in Figure 2.8 to ensure proper connection of all edges emanating from variable nodes at position  $t \in \{L - w, \dots, L - 1\}$  to a check node according to the connection rule defined above.

Including the boundary conditions for the randomized connections, the rate of a SC LDPC code ensemble is defined as follows.

**Definition 2.14** (Rate of a  $(J, K, L, w)$  LDPCC ensemble). *Given a  $(J, K, L, w)$  LDPCC ensemble with coupling length  $L$  and coupling width  $w$ , the rate is given as*

$$R = 1 - \frac{J}{K} \frac{L - w - 1 + 2 \sum_{i=0}^{w-1} \left(1 - \left(\frac{w-i-1}{w}\right)^K\right)}{L}. \quad (2.50)$$

The overhanging check nodes at positions  $t \in \{L, \dots, L + w - 2\}$ , unfortunately cause a rate-loss that vanishes linearly with the increase of  $L$ . In the limit as  $L \rightarrow \infty$  we get  $R = 1 - J/K$  which is the design rate of the uncoupled underlying regular LDPC ensemble. As we are most often concerned with very long coupled chains, the rate-loss is of minor importance for the remainder of the thesis. Additionally, due to the randomized connection

there is a remaining probability, that a check node in the range  $t \in \{L, \dots, L + w - 2\}$  is not connected to a variable node resulting in a rate increase. The vanishing influence of this effect with growing  $L$  gives rise to neglecting this behavior.

The irregular case is a straightforward generalization of the regular case where each edge connection is similarly distributed over a given coupling length  $w$ . More details and a definition of a  $(\lambda(x), \rho(x), L, w)$  LDPCC ensemble can be found in Section 4.3 and Definition 4.3. Similarly, the coupling procedure extends to the case of MET LDPCC ensembles. The only constraint to be taken into account is that the randomized edge connections are done per edge-type to ensure that an edge of type  $i$  emanating from a variable node at time instant  $t$  can only be connected to a check node in the range  $[t, t + w - 1]$  that accepts this edge type. Coupling of protograph ensembles is done in a similar way but here edges are not spread randomly over the coupling width but in a controlled manner. More details for this *edge spreading procedure* will be given in Section 5.2.

To assess the performance of a LDPCC code ensemble we introduce adapted DE equations to account for the connections within the SC chain. We again give an introduction with the help of a regular  $(J, K, L, w)$  LDPCC ensemble. Generalizations to irregular and MET ensembles are shortly noted for completeness.

To account for the code ensembles at positions  $t \in [0, L - 1]$  we define the density emitted from variable nodes at time instant  $t$  in iteration  $l$  as  $\mathbf{x}_t^{(l)}$ . Similarly, the density emitted from check nodes at position  $t$  is denoted as  $\mathbf{y}_t^{(l)}$ . Reconsider the DE recursion for regular block codes as

$$\mathbf{x}_t^{(l+1)} = \mathbf{a}_{BMS} \otimes \left( \mathbf{y}_t^{(l)} \right)^{\otimes (J-1)} \quad (2.51)$$

with  $\mathbf{y}_t^{(l)} = \left( \mathbf{x}_t^{(l)} \right)^{\boxtimes (K-1)}$ . This is the case for an uncoupled system which we can refer to as a coupled system with coupling width  $w = 1$ . As the coupling procedure starts to connect code ensembles at different time instants  $t$ , the density evolution equations have to account for the influence of densities of at time instants  $t' \neq t$  to the densities at time instant  $t$ . The coupling is done in a uniformly random fashion over a window of  $w$  time instants. Therefore, the resulting density that needs to be incorporated in the calculations at time instant  $t$  is simply an average of densities within the bespoke window of size  $w$ . To clarify, which densities shall be taken for average at check and variable node side, we define an average variable node density  $\bar{\mathbf{x}}_t^{(l)}$  in iteration  $l$  as

$$\bar{\mathbf{x}}_t^{(l)} = \frac{1}{w} \sum_{k=0}^{w-1} \mathbf{x}_{t-k}^{(l)} \quad (2.52)$$

and similarly the average density on the check node side  $\bar{\mathbf{y}}_t^{(l)}$  as

$$\bar{\mathbf{y}}_t^{(l)} = \frac{1}{w} \sum_{j=0}^{w-1} \mathbf{y}_{t+j}^{(l)}. \quad (2.53)$$

The average within these two quantities resembles exactly the SC connections from the ensemble definition as can be seen by the indexes. The definition of the DE recursion for a regular  $(J, K, L, w)$  LDPCC code ensemble is given in the following.

**Definition 2.15** (DE for a  $(J, K, L, w)$  LDPCC ensemble). *Given a  $(J, K, L, w)$  LDPCC ensemble, the density emitted by variable nodes in iteration  $l + 1$  is given by*

$$\mathbf{x}_t^{(l+1)} = \mathbf{c} \otimes \left( \frac{1}{w} \sum_{j=0}^{w-1} \left( \frac{1}{w} \sum_{k=0}^{w-1} \mathbf{x}_{t-k+j}^{(l)} \right)^{\boxtimes(K-1)} \right)^{\otimes(J-1)} \quad (2.54)$$

with  $\mathbf{c}$  as the  $L$ -density given by the respective channel.

In the above definition, we used the fact that

$$\mathbf{x}_t^{(l+1)} = \mathbf{c} \otimes \left( \bar{\mathbf{y}}_t^{(l)} \right)^{\otimes(J-1)} \quad (2.55)$$

and

$$\bar{\mathbf{y}}_t^{(l)} = \left( \bar{\mathbf{x}}_t^{(l)} \right)^{\boxtimes(K-1)} \quad (2.56)$$

together with (2.52) and (2.53).

The recursion can again be used to numerically calculate an iterative decoding threshold  $\xi^{\text{BP}}$  for the coupled ensemble. Note again, that the uncoupled case of block code ensembles is a special case of the coupled ensemble with  $w = 1$ .

The DE equations extend straightforward to the case for irregular and MET LDPCC ensembles. For the irregular case, the general DE recursion is given in Lemma 4.3. For the MET case we shortly state the DE recursion in form of a definition as follows.

**Definition 2.16** (DE of a  $(\nu(\underline{x}), \mu(\underline{x}), L, w)$  LDPCC ensemble). *Given a  $(\nu(\underline{x}), \mu(\underline{x}), L, w)$  LDPCC ensemble with coupling width  $w$  and coupling length  $L$ , the vector  $\underline{\mathbf{x}}_t^{(l)} = (\mathbf{x}_{t,1}^{(l)}, \mathbf{x}_{t,2}^{(l)}, \dots, \mathbf{x}_{t,m_e}^{(l)})$  represents the densities emitted from variable nodes at time instant  $t$  where  $\mathbf{x}_{t,i}^{(l)}$  denotes the density emitted from variable nodes of edge type  $i$  at time instant  $t$  in iteration  $l$ . With the slightly generalized definition of (2.52) and (2.53) given as*

$$\bar{\underline{\mathbf{x}}}_t^{(l)} = \frac{1}{w} \sum_{k=0}^{w-1} \underline{\mathbf{x}}_{t-k}^{(l)} \quad (2.57)$$

and

$$\bar{\underline{\mathbf{y}}}_t^{(l)} = \frac{1}{w} \sum_{j=0}^{w-1} \underline{\mathbf{y}}_{t+j}^{(l)} \quad (2.58)$$

the recursion for  $\underline{\mathbf{x}}_t^{(l)}$  is given as

$$\underline{\mathbf{x}}_t^{(l+1)} = \mathbf{c} \otimes \lambda \left( \frac{1}{w} \sum_{j=0}^{w-1} \rho \left( \frac{1}{w} \sum_{k=0}^{w-1} \underline{\mathbf{x}}_{t-k+j}^{(l)} \right) \right) \quad (2.59)$$

with  $\lambda(\underline{\mathbf{x}})$  and  $\rho(\underline{\mathbf{x}})$  from Definition 2.10.

The complexity of an LDPCC code ensemble does not differ from the complexity of LDPC code ensemble with same degrees and sufficiently large coupling length  $L$  as the

asymptotically, the variable degree distribution of uncoupled and coupled code ensembles are converging. Of interest for the complexity of LDPC code ensembles is the coupling width which directly influences the window size of a possible windowed BP decoding algorithm. In general, the smaller the coupling width  $w$ , the smaller the window size can be which is beneficial for decoder implementational complexity. We therefore always seek for smallest coupling width.

## Threshold saturation

Lentmaier et al. observed in [LF10] and [LSCZ10] that the iterative decoding thresholds of SC LDPC ensembles numerically coincide with the optimal maximum likelihood (ML) decoding threshold of the uncoupled LDPC ensemble. In [KRU11], the  $(J, K, L, w)$  ensemble was constructed to analytically investigate these findings with a mathematically tractable model. For the BEC it was shown that as

$$\lim_{w \rightarrow \infty} \lim_{L \rightarrow \infty} \lim_{n_t \rightarrow \infty} R(J, K, L, w) = 1 - \frac{J}{K} \quad (2.60)$$

which is the design rate of the underlying  $(J, K)$  LDPC block code ensemble. Additionally and remarkably,

$$\lim_{w \rightarrow \infty} \lim_{L \rightarrow \infty} \lim_{n_t \rightarrow \infty} \epsilon^{\text{BP}}(J, K, L, w) = \epsilon^{\text{MAP}}(J, K) \quad (2.61)$$

where  $\epsilon^{\text{MAP}}(J, K)$  is the *maximum a-posteriori (MAP)* threshold of the underlying  $(J, K)$  LDPC block code ensemble. If the degrees  $(J, K)$  are increased while keeping the ratio  $J/K$  constant to preserve the rate, the MAP threshold converges to the Shannon limit  $\epsilon^{\text{Sh}}$  as

$$\lim_{J \rightarrow \infty} \epsilon^{\text{MAP}} = \epsilon^{\text{Sh}}. \quad (2.62)$$

This two step argumentation leads to the fact that for the BEC, the Shannon limit could be achieved with regular  $(J, K)$  LDPC codes even with a sub-optimal decoder. In fact, the sub-optimal decoder in the uncoupled case becomes the optimal decoder in the coupled case.

Extensions to general channels were made in [KRU13] and [NYPN12]. Remarkably, this effect of threshold saturation through spatial coupling was shown to happen with a wide variety of ensembles such as protograph and MET ensembles. The idea of a threshold improvement through spatial coupling and its impact on low-complexity decoding is the main theme of this thesis.

## 2.8 Incremental Redundancy

The term *incremental redundancy (IR)* nowadays is mostly correlated with the use of HARQ protocols known from, e.g., LTE. Originally, two types of HARQ protocols were

introduced. Type 1 HARQ was simply the transmission of information bits that were encoded with a channel code  $\mathcal{C}$ . If the receiver was not able to decode the codeword a retransmission would be requested through a feedback channel. The problem of Type 1 HARQ was the inefficiency in high signal-to-noise ratio (SNR) situations. With the introduction of HARQ Type 2, IR was of utmost interest and first noted in [Man74]. Here the general notion was to only send parity bits of a codeword to the receiver if the channel quality was too bad to decode. The transmitter therefore subsequently sends IR packets of parity bits to the receiver as long as the receiver does not acknowledge the successful decoding. We give a rather simplistic definition of IR as follows.

**Definition 2.17** (Incremental Redundancy). *Separating a codeword from a code  $\mathcal{C}$  of length  $n$  into  $k$  bits representing the information to be transmitted (either systematic or not) and  $n - k$  bits representing the parity (redundancy), IR is defined as the procedure of dividing the  $n - k$  parity bits into subblocks of redundancy and, depending on its need, supplying one or a combination of redundancy subblocks in a subsequent fashion to the receiver (decoder). The cumulated bits at the receiver are then jointly used to decode the information.*

To support such a transmission scheme with IR, the channel code has to offer the capability to be decoded by only a subset of received blocks. Already [Man74] gave an exemplary construction of such a *rate-compatible* code design with Reed-Solomon Codes. In [Hag88], a rate-compatible coding scheme based on convolutional codes was introduced which showed significant gain to combat fading and increases throughput on Rayleigh fading channels. The rate-compatible code constraint was introduced by Hagenauer [Hag88] as follows.

**Definition 2.18** (Rate-compatible constraint [Hag88]). *All the code bits of a high rate code are used by the lower rate codes; or in other words, the high rate codes are embedded into the lower rate codes of the (rate-compatible) family.*

We seek to find code constructions that support this constructive constraint. For the convolutional codes in [Hag88], the technique of *puncturing* was introduced. A very low rate mother code is used to encode a codeword. To get codewords of higher rate a specific number of bits is simply masked (punctured) and not sent within the transmission of the codeword. The remaining codeword consisting of the unmasked bits has higher rate. On demand, the masked bits can be additionally send as IR for the decoder to be used. In modern communication systems, puncturing is used in conjunction with the more powerful Turbo Codes [BGT93]. The performance drawback of puncturing stems from the fact, that the mother code (of low rate) is highly optimized for good performance but as soon as the fraction of punctured bits increases (rate increases), the performance worsens. For the IR protocols, this is a disadvantage as the first transmission will be one of high rate and is therefore not as powerful as codewords at lower rates. An overcome to this issue,

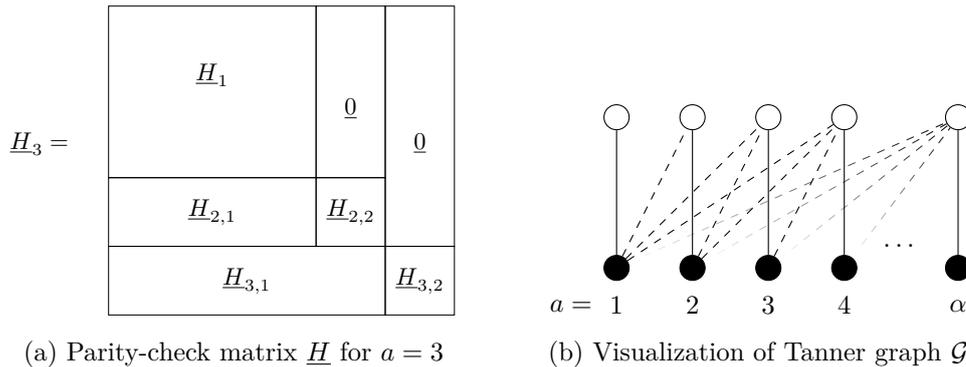


Figure 2.9: Parity-check matrix and Tanner graph representation of a rate-compatible code build through extension

the exact counterpart to puncturing called *extension*, will be the focus of this thesis. The procedure of extension starts with a high rate mother code  $\mathcal{C}$  with associated parity-check matrix  $\underline{H}$  and Tanner graph  $\mathcal{G}$  and subsequently appends rows and columns to  $\underline{H}$  (check and variable nodes in  $\mathcal{G}$ ) to generate new parity bits and lower the rate  $R$ . The structure of  $\underline{H}$  and  $\mathcal{G}$  is shown in Figure 2.9.

We construct a family of rate-compatible codes  $\{\mathcal{C}_a\}_{a=1}^\alpha$  where  $a$  denotes the index for the specific IR step and  $\alpha$  is the maximum number of members of the family and therefore IR steps. The corresponding parity-check matrix  $\underline{H}_a$  for  $a = 2, \dots, \alpha$  is given as

$$\underline{H}_a = \begin{bmatrix} \underline{H}_{a-1} & \underline{0} \\ \underline{H}_{a,1} & \underline{H}_{a,2} \end{bmatrix} \quad (2.63)$$

with  $\underline{H}_1$  as the parity-check matrix of the highest rate code. To support rate-compatibility, the extension is always carried out in such a way to ensure an all zero matrix in the upper right of the resulting parity check matrix. This is important as newly appended columns (redundant bits) do not alter check equations of previous IR steps. This nested structure can be rediscovered within the simplified Tanner graph visualization in Figure 2.9. At IR step  $a = 1$  just one variable and one check node denote the initial code (precode) with highest rate. Subsequent addition of check and variable nodes at each IR step are the corresponding action to adding rows and columns in the parity-check matrix. The rate-compatibility constraint that ensures the nested structure of the parity-check matrices can be seen easily. While connections from check nodes at IR step  $a$  are allowed to every variable node at IR step  $a' \leq a$ , variable nodes at step  $a$  are only allowed to have connections to check nodes at  $a' \geq a$ .

**Rateless Codes** The class of rateless codes form a specific subgroup of rate-compatible code constructions. These codes are characterized by a random approach to the creation of parity bits for subsequent IR steps. Using a vector  $\underline{v}$  of symbols (either binary or from larger fields), a rateless encoded symbol  $c_a$  at IR step  $a$  is generated as  $c_a = f(\underline{v})$  where

$f(\cdot)$  is a random function involving the use of a random number generator. The key to a good performing rateless code is the design of the random function  $f(\cdot)$  in a way, that the decoding algorithm does decode the correct codeword with high probability. Due to the random nature of the "on-the-fly" encoding, these code construction exhibit very good performance when the block length is chosen sufficiently large. A popular version of these rateless codes are Luby transform (LT) codes [Lub02], where the vector  $\underline{v}$  consists directly of the information symbols to be transmitted. Raptor codes [Sho06] are the second and more important variant of rateless codes where the vector  $\underline{v}$  is already encoded with an LDPC code as a precode. These codes are shown to achieve the Shannon limit on the BEC.

As rateless codes also produce IR, they possess the same principal structure as shown in Figure 2.9. The only difference is in the choice of the parity-check matrix which is random in every transmission for the rateless codes in contrast to the predetermined parity-check matrix of LDPC codes.

## 2.9 Summary

This chapter shall be the basic collection and description of tools and methods to be used throughout the rest of this thesis. In particular, the ultimate performance limit of Shannon was introduced. In the subsequent chapters, all code constructions will be compared against this limit. It is always of interest how close a specific code construction approaches the Shannon limit. Further, the three different models to define LDPC code ensembles are described, namely irregular, protograph based and MET LDPC ensembles. The specific terminology in terms of protograph matrices and degree distributions is stated based on different examples to visualize the differences and commonalities between these different description methodologies. A short discussion clarifies why we are interested in the analysis of ensembles instead of specific codes. The tool to analyze and compare the performance against the Shannon limit is also introduced for all types of ensembles that we use throughout the thesis. Finally, a short wrap up of the basics for the two main ingredients of this thesis are discussed. The introduction of LDPC codes as well as a short explanation of IR and rate-compatible code constructions. These two topics describe the main focus of this thesis and their interplay, issues and performance are the focus of the following chapters.

## 2.10 Related Literature

The channel models considered in this thesis are rather simple but important ones taken from coding and information theory. Their description together with extensive discussions about their channel capacity can be found in various textbooks on information theory

such as [CT06], [Gal68] and [Fri96]. An in depth introduction to LDPC codes can be found in various books such as [RU08], [RL09] and [LC04]. The notion of irregular LDPC ensembles together with the asymptotic analysis with DE was introduced in [LMSS98, LMSS01a, LMSS01b]. The MET paradigm and model was in detail presented and analyzed in [RU04] and additionally described in [RU08]. The BP algorithm was first introduced in [Pea88] without any relation to the decoding of channel codes. The decoding algorithms in Gallagers PhD thesis [Gal63] were shown to be instances of this algorithm. The direct application of Pearls BP algorithm to decoding of LDPC codes was discussed in [MN95]. A specific decoding complexity metric for the BP decoder was introduced in [MCFF10] for measuring the complexity only based on the code construction. The first mentioning of LDPCC codes was given in [JFZ99]. A recent overview of the topic can be found in [CDF<sup>+</sup>14]. The topic of IR was first introduced in [Man74] in conjunction with a concrete coding scheme. Later, the term rate-compatibility was introduced for a specific code design constraint for IR in [Hag88]. Rate-compatibility through puncturing for LDPCC codes was investigated in [ZMGC13] but the investigations were not directed towards the general iterative decoding behavior as only simulations of specific code constructions were shown. Rate-compatible extension for LDPCC codes will be discussed in Section 5 for a literature overview. SC LT codes were investigated in [AU11], where the robustness increasing effect of spatial-coupling helps to get outstanding asymptotic performance on a variety of different channel models. In [HXR12], the focus of the investigation was drawn to the different options of SC precode and/or SC LT code. In [SKS13] and [SKS14] the authors gave the analytical assessment to the capacity achieving behavior of SC Raptor codes.