

# CHALMERS



## ANALYSIS AND DESIGN OF SPATIALLY-COUPLED CODES WITH APPLICATION TO FIBER-OPTICAL COMMUNICATIONS

CHRISTIAN HÄGER

*Communication Systems Group*  
*Department of Signals and Systems*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden, 2016



THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

---

Analysis and Design of Spatially-Coupled  
Codes with Application to Fiber-Optical  
Communications

CHRISTIAN HÄGER



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Communication Systems Group  
Department of Signals and Systems  
Chalmers University of Technology  
Göteborg, Sweden, 2016

**Analysis and Design of Spatially-Coupled  
Codes with Application to Fiber-Optical  
Communications**

CHRISTIAN HÄGER

Copyright © 2016 CHRISTIAN HÄGER, except where  
otherwise stated. All rights reserved.

ISBN 978-91-7597-351-7  
Doktorsavhandlingar vid Chalmers tekniska högskola  
Series No. 4032  
ISSN 0346-718X

This thesis has been prepared using L<sup>A</sup>T<sub>E</sub>X and PSTricks.

Communication Systems Group  
Department of Signals and Systems  
Chalmers University of Technology  
SE-412 96 Göteborg, Sweden  
Phone: +46 (0)31 772 1000  
[www.chalmers.se](http://www.chalmers.se)

Printed by Chalmers Reproservice  
Göteborg, Sweden, May 2016

## Abstract

The theme of this thesis is the analysis and design of error-correcting codes that are suitable for high-speed fiber-optical communication systems. In particular, we consider two code classes. The codes in the first class are protograph-based low-density parity-check (LDPC) codes which are decoded using iterative soft-decision decoding. The codes in the second class are generalized LDPC codes with degree-2 variable nodes—henceforth referred to as generalized product codes (GPCs)—which are decoded using iterative bounded-distance decoding (BDD). Within each class, our focus is primarily on spatially-coupled codes. Spatially-coupled codes possess a convolutional structure and are characterized by a wave-like decoding behavior caused by a termination boundary effect. The contributions of this thesis can then be categorized into two topics, as outlined below.

First, we consider the design of systems operating at high spectral efficiency. In particular, we study the optimization of the mapping of the coded bits to the modulation bits for a polarization-multiplexed system that is based on the bit-interleaved coded modulation paradigm. As an example, for the (protograph-based) AR4JA code family, the transmission reach can be extended by roughly up to 8% by using an optimized bit mapper, without significantly increasing the system complexity. For terminated spatially-coupled codes with long spatial length, the bit mapper optimization only results in marginal performance improvements, suggesting that a sequential allocation is close to optimal. On the other hand, an optimized allocation can significantly improve the performance of tail-biting spatially-coupled codes which do not possess an inherent termination boundary. In this case, the unequal error protection offered by the modulation bits of a nonbinary signal constellation can be exploited to create an artificial termination boundary that induces a wave-like decoding for tail-biting spatially-coupled codes.

As a second topic, we study deterministically constructed GPCs. GPCs are particularly suited for high-speed applications such as optical communications due to the significantly reduced decoding complexity of iterative BDD compared to iterative soft-decision decoding of LDPC codes. We propose a code construction for GPCs which is sufficiently general to recover several well-known classes of GPCs as special cases, e.g., irregular product codes (PCs), block-wise braided codes, and staircase codes. Assuming transmission over the binary erasure channel, it is shown that the asymptotic performance of the resulting codes can be analyzed by means of a recursive density evolution (DE) equation. The DE analysis is then applied to study three different classes of GPCs: spatially-coupled PCs, symmetric GPCs, and GPCs based on component code mixtures.

**Keywords:** Bit-interleaved coded modulation, bit mapper, bounded-distance decoding, braided codes, density evolution, generalized low-density parity-check codes, generalized product codes, spatial coupling, staircase codes.



# List of Publications

This thesis is based on the following publications:

## Paper A

C. Häger, A. Graell i Amat, F. Brännström, A. Alvarado, and E. Agrell, “Improving Soft FEC Performance for Higher-Order Modulations via Optimized Bit Channel Mappings,” *Optics Express*, vol. 22, no. 12, pp. 14544–14558, June 2014.

## Paper B

C. Häger, A. Graell i Amat, A. Alvarado, F. Brännström, and E. Agrell, “Terminated and Tailbiting Spatially Coupled LDPC Codes with Optimized Bit Mapping for Spectrally Efficient Fiber-Optical Systems,” (invited paper) *J. Lightw. Technol.*, vol. 33, no. 7, pp. 1275–1285, April 2015.

## Paper C

C. Häger, A. Graell i Amat, H. D. Pfister, A. Alvarado, F. Brännström, and E. Agrell, “On Parameter Optimization for Staircase Codes,” in *Proc. Optical Fiber Communication Conf. and Exposition (OFC)*, Los Angeles, CA, 2015.

## Paper D

C. Häger, H. D. Pfister, A. Graell i Amat, and F. Brännström, “Density Evolution for Deterministic Generalized Product Codes on the Binary Erasure Channel,” submitted to *IEEE Trans. Inf. Theory*.

## Paper E

C. Häger, H. D. Pfister, A. Graell i Amat, and F. Brännström, “Density Evolution and Error Floor Analysis for Staircase and Braided Codes,” in *Proc. Optical Fiber Communication Conf. and Exposition (OFC)*, Los Angeles, CA, 2016.

## Paper F

C. Häger, H. D. Pfister, A. Graell i Amat, and F. Brännström, “Deterministic and Ensemble-Based Spatially-Coupled Product Codes,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Barcelona, Spain, 2016.

Other publications by the author not included in this thesis:

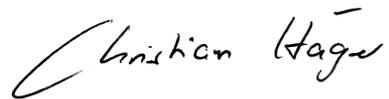
- C. Häger, A. Graell i Amat, A. Alvarado, and E. Agrell, “Constellation Optimization for Coherent Optical Channels Distorted by Nonlinear Phase Noise,” in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, Anaheim, CA, 2012.
- C. Häger, A. Graell i Amat, A. Alvarado, and E. Agrell, “Design of APSK Constellations for Coherent Optical Channels with Nonlinear Phase Noise,” *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3362–3373, Aug. 2013.
- C. Häger, L. Beygi, E. Agrell, P. Johansson, M. Karlsson, and A. Graell i Amat, “A Low-Complexity Detector for Memoryless Polarization-Multiplexed Fiber-Optical Channels,” *IEEE Commun. Lett.*, vol. 18, no. 2, pp. 368–371, Jan. 2014.
- C. Häger, A. Graell i Amat, A. Alvarado, F. Brännström, and E. Agrell, “Optimized Bit Mappings for Spatially Coupled LDPC Codes over Parallel Binary Erasure Channels,” in *Proc. IEEE Int. Conf. Communications (ICC)*, Sydney, Australia, 2014.
- C. Häger, A. Graell i Amat, F. Brännström, A. Alvarado, and E. Agrell, “Comparison of Terminated and Tailbiting Spatially Coupled LDPC Codes With Optimized Bit Mapping for PM-64-QAM,” in *Proc. European Conf. Optical Communication (ECOC)*, Cannes, France, 2014.
- A. Graell i Amat, C. Häger, F. Brännström, and E. Agrell, “Spatially-Coupled Codes for Optical Communications: State-of-the-Art and Open Problems,” (invited paper) in *Proc. Optoelectronics and Communications Conf. (OECC)*, Shanghai, China, 2015.
- M. Ivanov, C. Häger, F. Brännström, A. Graell i Amat, A. Alvarado, and E. Agrell, “On the Information Loss of the Max-Log Approximation in BICM Systems,” *IEEE Trans. Inf. Theory* (to appear), 2016.
- C. Häger, H. D. Pfister, A. Graell i Amat, F. Brännström, and E. Agrell, “A Deterministic Construction and Density Evolution Analysis for Generalized Product Codes,” (invited paper) in *Proc. Int. Zurich Seminar on Communications*, Zürich, Switzerland, 2016.
- C. Häger, A. Graell i Amat, H. D. Pfister, and F. Brännström, “Density Evolution for Deterministic Generalized Product Codes with Higher-Order Modulation,” (invited paper) in *Proc. Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, 2016.

## Acknowledgements

First of all, I would like to express my deepest gratitude to my doctoral advisor (*Doktorvater* in German) Professor Alexandre Graell i Amat. Àlex has been a constant source of support and guidance during my studies and I am very grateful for the invaluable feedback that I have received from him throughout the past four and a half years. I would also like to express my gratitude to my co-advisor Professor Fredrik Brännström for all the fruitful discussions and for spotting even the slightest inconsistencies in my paper drafts. I would further like to thank Dr. Alex Alvarado and Professor Erik Agrell who have been my co-advisors until April 2014. Alex has also been an excellent host during my research visit to the University of Cambridge in 2013 for which I am very grateful.

I am especially indebted to Professor Henry Pfister who has had a great influence on the shape and direction of my doctoral studies. I had the pleasure to visit Henry for the first time in autumn 2014, where he suggested to work on the asymptotic analysis of deterministic codes. At least half of the work contained in this thesis directly originates from this collaboration which is still ongoing.

I would like to extend my thanks to Professor Iryna Andriyanova for sharing her insight on spatially-coupled LDPC codes during my stay in Paris. Also, one of the persons who taught me a lot during my “early” PhD years is Dr. Lotfollah Beygi. I am immensely grateful for all the time that he spent with me discussing various aspects of fiber-optical communications. I would also like to thank Mikhail Ivanov for being an outstanding office mate, research collaborator, and friend. Many thanks also to the administrative staff of the department, in particular Agneta Kinnander, Madeleine Persson, and Natasha Adler. Special thanks goes to the late Lars Börjesson, who will be sorely missed as a computer administrator and colleague. Last but not least, I would like to thank everyone in our group, and also everyone involved in FORCE, the fiber-optic research center at Chalmers, for creating a fantastic and stimulating work environment.



Christian Häger  
Göteborg, April 2016

*Regarding the financial support, I would like to thank and acknowledge the Ericsson Research Foundation for their generous financial support of my research visits to Cambridge and Duke University. Furthermore, this work was partially funded by the Swedish Research Council under grant #2011-5961.*



# Acronyms

The following acronyms are used in the introductory part of this thesis.

Abbreviation	Meaning
<b>ASE</b>	amplified spontaneous emission
<b>AWGN</b>	additive white Gaussian noise
<b>BCH</b>	Bose–Chaudhuri–Hocquenghem
<b>BDD</b>	bounded-distance decoding
<b>BEC</b>	binary erasure channel
<b>BICM</b>	bit-interleaved coded modulation
<b>BP</b>	belief propagation
<b>BSC</b>	binary symmetric channel
<b>CM</b>	coded modulation
<b>CN</b>	constraint node
<b>DE</b>	density evolution
<b>EDFA</b>	erbium-doped fiber amplifier
<b>EXIT</b>	extrinsic information transfer
<b>GLDPC</b>	generalized low-density parity-check
<b>GPC</b>	generalized product code
<b>HPC</b>	half-product code
<b>LDPC</b>	low-density parity-check
<b>LLR</b>	log-likelihood ratio
<b>MI</b>	mutual information
<b>NLSE</b>	nonlinear Schrödinger equation
<b>OOK</b>	on-off keying
<b>PC</b>	product code
<b>PDF</b>	probability density function
<b>PM</b>	polarization-multiplexed
<b>PSD</b>	power spectral density
<b>QPSK</b>	quadrature phase-shift keying
<b>SMF</b>	single-mode fiber
<b>sNLSE</b>	stochastic nonlinear Schrödinger equation
<b>SNR</b>	signal-to-noise ratio
<b>SSFM</b>	split-step Fourier method
<b>VN</b>	variable node



---

# Contents

---

<b>Abstract</b>	<b>i</b>
<b>List of Papers</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Acronyms</b>	<b>vii</b>
<b>I Introduction</b>	<b>1</b>
<b>1 Background</b>	<b>3</b>
1.1 Thesis Organization . . . . .	6
1.2 Notation . . . . .	6
<b>2 Fiber-Optical Channel Modeling</b>	<b>9</b>
2.1 The Additive White Gaussian Noise Channel . . . . .	10
2.2 The Nonlinear Schrödinger Equation . . . . .	10
2.3 Optical Amplification and Noise . . . . .	12
2.4 Polarization Multiplexing . . . . .	14
2.5 Linear Pulse Modulation and Linear Receiver . . . . .	15
<b>3 Bit-Interleaved Coded Modulation</b>	<b>17</b>
3.1 Introduction to Coded Modulation . . . . .	18
3.2 BICM System Model . . . . .	19

<b>4</b>	<b>Low-Density Parity-Check Codes</b>	<b>23</b>
4.1	Introduction . . . . .	24
4.2	Iterative Belief Propagation Decoding . . . . .	24
4.3	Code Construction via Protographs . . . . .	26
4.4	Density Evolution . . . . .	27
4.5	Spatially-Coupled LDPC Codes . . . . .	29
<b>5</b>	<b>Generalized Product Codes</b>	<b>33</b>
5.1	Generalized Low-Density Parity-Check Codes . . . . .	34
5.2	Product Codes . . . . .	34
5.3	Generalized Product Codes . . . . .	36
5.3.1	Braided Codes . . . . .	36
5.3.2	Staircase Codes . . . . .	37
5.3.3	Half-Product Codes . . . . .	38
5.4	Iterative Bounded-Distance Decoding . . . . .	39
5.5	Density Evolution . . . . .	39
5.5.1	Code Ensembles . . . . .	40
5.5.2	Deterministic Codes . . . . .	41
<b>6</b>	<b>Conclusions and Future Work</b>	<b>43</b>
	<b>Bibliography</b>	<b>47</b>
<b>II</b>	<b>Papers</b>	<b>53</b>
<b>A</b>	<b>Improving Soft FEC Performance for Higher-Order Modulations via Optimized Bit Channel Mappings</b>	<b>A1</b>
1	Introduction . . . . .	A3
1.1	Notation . . . . .	A5
2	System Model . . . . .	A6
2.1	Continuous-Time Channel . . . . .	A6
2.2	Discrete-Time Channel . . . . .	A6
2.3	Bit-Interleaved Coded Modulation . . . . .	A7
3	Protograph-Based LDPC Codes . . . . .	A9
3.1	AR4JA Codes . . . . .	A10
3.2	Spatially Coupled LDPC Codes . . . . .	A10
3.3	Decoding and Asymptotic EXIT Analysis . . . . .	A10
4	Bit Mapper Optimization . . . . .	A12
4.1	Asymptotic Bit Mapper Model . . . . .	A12
4.2	Optimization . . . . .	A13
5	Results and Discussion . . . . .	A15

5.1	Linear Transmission . . . . .	A16
5.2	Nonlinear Transmission . . . . .	A17
6	Conclusion . . . . .	A19
	References . . . . .	A19

**B Terminated and Tailbiting Spatially-Coupled Codes with Optimized Bit Mappings for Spectrally Efficient Fiber-Optical Systems** **B1**

1	Introduction . . . . .	B3
2	System Model . . . . .	B5
3	Protograph-Based SC-LDPC Codes . . . . .	B7
	3.1 Code Construction . . . . .	B7
	3.2 Soft-Decision Decoding and Density Evolution . . . . .	B8
4	SC-GLDPC Codes with BCH Component Codes . . . . .	B10
	4.1 Code Construction . . . . .	B10
	4.2 Hard-Decision Decoding and Density Evolution . . . . .	B12
5	Bit Mapper Optimization . . . . .	B14
6	Results and Discussion . . . . .	B14
	6.1 Structure of the Optimized Bit Mapper for Tailbiting Codes . . . . .	B15
	6.2 Optimization Gain . . . . .	B16
	6.3 Gap to Capacity . . . . .	B18
	6.4 Simulation Results . . . . .	B19
7	Conclusions . . . . .	B20
8	Acknowledgements . . . . .	B21
	References . . . . .	B21

**C On Parameter Optimization for Staircase Codes** **C1**

1	Introduction . . . . .	C3
2	Staircase Codes . . . . .	C4
	2.1 Iterative Hard-Decision Decoding . . . . .	C4
	2.2 Density Evolution . . . . .	C4
3	Extended Staircase Codes . . . . .	C6
4	Results and Discussion . . . . .	C6
5	Conclusions . . . . .	C8
	References . . . . .	C8

**D Density Evolution for Deterministic Generalized Product Codes on the Binary Erasure Channel** **D1**

1	Introduction . . . . .	D3
	1.1 Notation . . . . .	D6
2	Half-Product Codes . . . . .	D6
	2.1 Code Construction . . . . .	D6
	2.2 Binary Erasure Channel . . . . .	D8

2.3	Iterative Decoding . . . . .	D8
2.4	Asymptotic Performance . . . . .	D9
3	Random Graphs and Branching Processes . . . . .	D12
3.1	Random Graphs . . . . .	D12
3.2	Neighborhood Exploration Process . . . . .	D13
3.3	Branching Processes . . . . .	D15
4	Proof of Theorem 1 . . . . .	D15
4.1	Tree-like Neighborhood . . . . .	D16
4.2	Convergence to the Poisson Branching Process . . . . .	D17
4.3	Density Evolution . . . . .	D18
4.4	Concentration . . . . .	D19
5	Generalized Product Codes . . . . .	D21
5.1	Code Construction . . . . .	D21
5.2	Inhomogeneous Random Graphs . . . . .	D25
5.3	Iterative Decoding . . . . .	D27
5.4	Asymptotic Performance . . . . .	D27
5.5	Density Evolution . . . . .	D29
6	Discussion . . . . .	D30
6.1	Thresholds and Code Comparisons . . . . .	D30
6.2	Upper Bound on the Decoding Threshold . . . . .	D31
6.3	Modified Decoding Schedules . . . . .	D31
6.4	Performance on the Binary Symmetric Channel . . . . .	D32
7	Irregular Half-Product Codes . . . . .	D33
7.1	Preliminaries . . . . .	D33
7.2	Lower Bounds on the Threshold . . . . .	D34
7.3	Optimization via Linear Programming . . . . .	D35
7.4	Initial Component Code Loss . . . . .	D36
7.5	Simulation Results . . . . .	D40
8	Conclusions and Future Work . . . . .	D42
A	Proof of Lemma 1 . . . . .	D42
B	Bound on the Second Moment of $T_\ell$ . . . . .	D43
	References . . . . .	D44

**E Density Evolution and Error Floor Analysis for Staircase and Braided Codes E1**

1	Introduction . . . . .	E3
2	Density Evolution for Deterministic GPCs . . . . .	E3
2.1	Code Construction . . . . .	E4
2.2	Iterative Decoding . . . . .	E4
2.3	Density Evolution . . . . .	E4
3	Case Study: Comparison of Staircase, Braided, and Half-Braided Codes . . . . .	E5
3.1	Parameters and Error Floor . . . . .	E6

3.2	Results and Discussion . . . . .	E7
4	Conclusion . . . . .	E8
	References . . . . .	E8

<b>F</b>	<b>Deterministic and Ensemble-Based Spatially-Coupled Product Codes</b>	<b>F1</b>
1	Introduction . . . . .	F3
2	Code Construction and Density Evolution for Deterministic Generalized Product Codes . . . . .	F4
2.1	Code Construction . . . . .	F4
2.2	Iterative Decoding . . . . .	F5
2.3	Density Evolution . . . . .	F5
3	Spatially-Coupled Product Codes . . . . .	F6
3.1	Deterministic Spatially-Coupled Product Codes . . . . .	F6
3.2	Spatially-Coupled Product Code Ensembles . . . . .	F7
4	Comparison of Deterministic and Ensemble-Based Codes . . . . .	F8
4.1	Ensemble Performance via Deterministic Codes . . . . .	F9
4.2	Simpler Deterministic Codes . . . . .	F10
5	Potential Threshold Optimization . . . . .	F11
6	Conclusion . . . . .	F11
A	Proof of Theorems 3 and 4 . . . . .	F12
	References . . . . .	F13



# **Part I**

## **Introduction**



# CHAPTER 1

---

## Background

---

When requesting a website, most internet users are probably unaware that the digital data is modulated onto a light source and transmitted over thousands of kilometers in an optical waveguide, a so-called optical fiber, at some point on the way from the remote server to their home computer or mobile device. In fact, more than 99% of the global intercontinental traffic is carried over optical fiber and such “long-haul” fiber-optical communication systems are the key enabler of high-speed internet data transfer connecting cities, countries, and continents [1].

There is currently a significant interest in determining the ultimate capacity limits of fiber-optical systems [2–4] and developing practical schemes that can achieve these limits [5–7]. Error-correcting codes are an integral part of communication systems that operate close to capacity. In theory, the proper use of such codes allows the system to achieve an arbitrarily low error rate if the data rate is chosen below the capacity [8]. In practice, however, operating at lower error rates and closer to capacity comes at the expense of an increased system complexity and communication delay. Code design thus requires assessment of nontrivial trade-offs between performance, complexity, and delay.

Fiber-optical communication systems operate at very high data rates that can exceed several hundreds of Gbit/s. At such high speeds, one of the main challenges is to keep the decoding complexity at an acceptable level. For example, the amount of processing power that can be spent on decoding each bit is severely limited [6]. Fiber-optical communication systems also require extremely low error rates below  $10^{-15}$  [6]. On the other hand, communication delay caused by coding is typically not an issue. This implies that the use of codes with long block lengths is relatively unproblematic. As an example, the code proposals in [9, 10] have effective block lengths in the order of  $10^6$  bits. Assuming

a transmission rate of 100 Gbit/s, the corresponding delay is then only  $20\ \mu\text{s}$ . This is negligible compared to the propagation delay caused by the light traversing hundreds or even thousands of kilometers of fiber.<sup>1</sup> In a nutshell, the challenge is therefore to devise coding schemes that offer very low error rates with affordable decoding complexity, while potentially using relatively long block lengths.

Current state-of-the-art codes are defined on graphs and decoded iteratively by passing messages along the edges of the graph [11, 12]. In this thesis, our focus is primarily on spatially-coupled codes. The first instance of spatially-coupled codes are the low-density parity-check (LDPC) convolutional codes introduced in [13], which are now also referred to as spatially-coupled LDPC codes. It was later realized in [14] that these codes have much better performance than conventional regular LDPC codes [15]. The reason for this phenomenon is a wave-like decoding behavior caused by a termination boundary effect. Several different proofs now exist for the fact that spatially-coupled LDPC codes can operate arbitrarily close to the capacity of a variety of different communication channels [16, 17]. Other instances of spatially-coupled codes include braided codes [18] and staircase codes [9], which have been shown to offer outstanding performance using very low-complexity decoding algorithms. In this thesis, we address several challenges that arise in the analysis and design of spatially-coupled codes when used in fiber-optical communication systems.

We start by studying the design of spectrally-efficient systems. Fiber-optical systems traditionally employ digital modulation techniques that are rather wasteful with the available frequency spectrum. As an example, switching the light source on and off according to the digital data stream—referred to as on-off keying (OOK)—is highly inefficient from a spectral viewpoint. To keep up with the increasing data rate demands of current applications, and to enable innovative broadband technologies in the future, it becomes more and more apparent that next-generation fiber-optical systems need to use the available spectrum more efficiently. To improve the spectral efficiency over OOK, the data can be encoded into multiple amplitude and/or phase levels of the optical carrier. A further increase in spectral efficiency can be achieved by utilizing both polarizations of the optical light, which is referred to as polarization-multiplexed (PM) transmission. PM signals can be represented as points in a four-dimensional signal space, also referred to as a signal constellation [19]. In Papers A and B, we study how to combine such signal constellations with error-correcting codes. In Paper A, we consider LDPC codes which are defined via protographs [20]. Protograph-based LDPC codes allow for an efficient hardware implementation, which makes them attractive candidates for high-speed application such as fiber-optical communications [6].

Spatially-coupled LDPC codes can also be constructed using protographs. An in-depth analysis focusing exclusively on spatially-coupled LDPC codes for spectrally-efficient systems is presented in Paper B. In order to highlight one of the contributions of this thesis, we first note that the termination boundary that induces the wave-like decoding behav-

---

<sup>1</sup>As a rule of thumb, light takes approximately 5 ms to propagate through 1000 kilometer of fiber.

---

ior for spatially-coupled LDPC codes comes at the price of a so-called rate loss, i.e., a larger redundancy overhead, compared to “uncoupled” regular LDPC codes. This rate loss can be avoided by considering a tail-biting termination scheme. The resulting codes are referred to as tail-biting spatially-coupled LDPC codes. Unfortunately, by default, these codes behave essentially the same as regular LDPC codes because the absence of an explicit termination boundary prevents a wave-like decoding behavior. One of the main findings of this work is that the different modulation bits of a nonbinary signal constellation can be exploited to create an artificial termination boundary. This artificial boundary is sufficient to initiate a decoding wave and can significantly improve the performance of “rate-loss-free” tail-biting spatially-coupled LDPC codes.

As a second topic, we investigate so-called generalized product codes (GPCs). GPCs are extensions of classical product codes (PCs) [21]. PCs are one of the first examples of the idea to build long and powerful codes from shorter component codes. In particular, each coded bit in a PC is protected by two component codes, where the coded bits are assumed to be arranged in a rectangular array. This assumption is relaxed for GPCs which allows the array shape to be arbitrary. GPCs are very appealing for high-speed applications such as fiber-optical communication systems. The reason is rooted in their low-complexity decoding algorithm, which is based on iteratively decoding the component codes. When compared to message-passing decoding of LDPC codes, this approach can result in significant complexity advantages [9]. Indeed, PCs are already implemented in certain communication standards for fiber-optical systems [22]. Moreover, several constructions of GPCs, e.g., braided [10] and staircase codes [9], have been recently proposed and investigated for such systems.

GPCs are the main focus in Papers C–F (although we already consider the application of GPCs for the bit mapper optimization in Paper B as a side application). In Paper C, we study parameter optimization for staircase codes. This work is inspired by the work in [23], where staircase code parameters are found using a simulation-based approach. The parameter optimization in Paper C on the other hand is based on density evolution (DE). DE is an analytical tool to analyze the behavior of codes under iterative decoding in the limit of infinite block lengths [24]. However, the DE analysis used in Paper C does not directly apply to staircase codes. Rather, we observed that staircase codes share some structural similarities with the spatially-coupled PC ensemble defined in [25]. A code ensemble is a collection or set of codes, typically defined via suitable randomized connections in the underlying graphical representation. The approach used in Paper C is therefore only heuristically motivated. While it appears to work well, it raises the question whether an asymptotic DE analysis is possible by directly targeting specific deterministic GPCs such as staircase codes.

This question is answered positively in Paper D, which represents the main theoretical contribution of this thesis. In Paper D, we propose a parametrized family of deterministic GPCs that includes staircase codes (and also many other code classes) as special cases. Based on the theory of inhomogeneous random graphs [26], we provide a DE analysis that

characterizes the asymptotic code performance. It is important to stress that the analysis does not rely on the definition of a code ensemble but directly applies to sequences of deterministically constructed codes. The resulting DE analysis can be useful for a variety of different applications. For example, it can be used to predict and compare the waterfall performance of different GPCs, optimize code parameters for particular classes of GPCs, find suitable windowed-decoding schedules for spatially-coupled PCs, or design new classes of GPCs. Papers E and F are based on the theoretical tools derived in Paper D, where we apply the theory to design and study deterministically constructed GPCs in more detail.

## 1.1 Thesis Organization

The format of this thesis is a so-called *collection of papers*. It is divided into two parts, where the first part serves as an introduction to the appended papers in the second part.

The remainder of the introductory part of this thesis is structured as follows. In Chapter 2, we provide an introduction to fiber-optical channel modeling and describe the origin of the channel models that are used in the appended papers. In Chapter 3, we give a brief introduction to bit-interleaved coded modulation (BICM), which is a pragmatic way to combine signal constellations with error-correcting codes to operate at high spectral efficiencies. In Chapter 4, we review some basic background about LDPC codes. In particular, we discuss iterative belief propagation (BP) decoding, DE, and protograph-based constructions including spatially-coupled LDPC codes. The content of this chapter is mainly relevant for Papers A and B, where the reader is assumed to be somewhat familiar with LDPC codes and iterative decoding. In Chapter 5, we then discuss GPCs. Starting from the concept of generalized low-density parity-check (GLDPC) codes, we give several examples of well-known classes of GPCs and discuss the decoding via iterative bounded-distance decoding (BDD). We also briefly review two approaches to perform an asymptotic analysis for GPCs. Finally, the main conclusions from the appended papers are summarized in Chapter 6, where we also discuss future work.

## 1.2 Notation

The following notation is used in the introductory part of this thesis.

- Vectors and matrices are typeset in bold font by lowercase letters  $\mathbf{a}$  and capital letters  $\mathbf{A}$ , respectively.
- The transpose of a matrix is denoted by  $(\cdot)^\top$ .
- $\mathbb{Z}$ ,  $\mathbb{N}_0$ ,  $\mathbb{N}$ ,  $\mathbb{R}$ , and  $\mathbb{C}$  denote the set of integers, nonnegative integers including zero, nonnegative integers excluding zero, real numbers, and complex numbers,

respectively.

- The cardinality of a set  $\mathcal{A}$  is denoted by  $|\mathcal{A}|$ .
- Random variables are denoted by capital letters  $X$  and their realizations by lower-case letters  $x$ .
- The probability density function (PDF) of a random variable  $X$  is denoted by  $f_X(x)$ . The PDF of a random variable  $Y$  conditioned on the realization of another random variable  $X$  is denoted by  $f_{Y|X}(y|x)$ .
- Expectation is denoted by  $\mathbb{E}[\cdot]$ .
- $\delta(t)$  denotes Dirac's delta function while  $\delta[k]$  denotes the Kronecker delta.
- Convolution is denoted by  $\otimes$ .
- The imaginary unit is denoted by  $j \triangleq \sqrt{-1}$ .
- Complex conjugation is denoted by  $(\cdot)^*$ .

### Notational Inconsistencies

The reader should be aware of the following inconsistencies in the notation across the appended papers and the thesis introduction.

- The variable  $n$  is used to denote the block length of an LDPC code and GLDPC code in Papers A, B, and the thesis introduction. It is used for different purposes in Papers C–F, e.g., it denotes the length of a Bose–Chaudhuri–Hocquenghem (BCH) code in Paper C.
- The block size of staircase codes and braided codes is denoted differently in different papers and the thesis introduction. For example, for staircase codes, the block size is denoted by  $a$  in Paper C, E and the thesis introduction, while it is denoted by  $d$  in Paper F. In Paper D, the block size is denoted by  $n_i$ , where the index  $i$  indicates the position in the Tanner graph.
- The error-correcting capability of a component code is denoted by  $t$  in Papers B, C, F, and the thesis introduction, while it is denoted by  $\mathbf{t}$  in Papers D and E. The variable  $t$  is also used as a time index for the signals in Papers A, B, and Chapter 2 of the thesis introduction.
- The matrix  $\boldsymbol{\eta}$  that defines the Tanner graph connectivity of the deterministic GPC construction is typeset in normal weight font as  $\eta$  in Paper E.
- The two light polarizations are denoted by  $\mathbf{a}$  and  $\mathbf{b}$  in the thesis introduction, while we use  $\mathbf{x}$  and  $\mathbf{y}$  in Papers A and B.

- The acronym “SC” is used for “spatially-coupled” in Papers A, B, and F, while it is used for “staircase code” in Paper E.
- The acronym “CN” is used for “constraint node” in Papers D, E, F, and the thesis introduction, while it is used for “check node” in Papers A and B.
- The spatially-coupled code ensemble in [27] is referred to as a “SC-GLDPC ensemble” in Papers B and C, while it is referred to as a “spatially-coupled PC ensemble” in Paper F and the thesis introduction.

---

## Fiber-Optical Channel Modeling

---

A channel model is a mathematical description of the propagation medium and possibly also includes certain elements of the transmitter and receiver (e.g., filters). In the appended papers, we assume “traditional” channel models, in particular the (discrete and memoryless) additive white Gaussian noise (AWGN) channel. This chapter is intended to put this channel model into the context of fiber-optical communications. It should also give the reader a broader picture about optical channel modeling in general. Other channel models that are mainly relevant for iterative hard-decision decoding are discussed in Chapter 5.

We are concerned with coherent, long-haul (i.e., distances exceeding 2000 km) data transmission over single-mode fibers (SMFs). The main challenge from a channel modeling perspective is a nonlinear effect caused by the relatively high signal power in relation to the small cross-section area of the fiber. Without going further into the physical details, a useful way to think about this effect is to imagine that the presence of an optical signal can compress the fiber material (in most cases silica) to such a degree that its propagation properties, in particular the refractive index, are changed in a nonlinear way [28, p. 18].

The chapter is structured as follows. In Section 2.1, we review the AWGN channel model. In Section 2.2, we discuss the nonlinear Schrödinger equation (NLSE) which is a deterministic channel model for an SMF. Multi-span links consisting of several SMFs including optical amplification elements are covered in Section 2.3. PM systems are discussed in 2.4. Lastly, linear modulation and receivers are described in Section 2.5.

## 2.1 The Additive White Gaussian Noise Channel

Consider the discrete and memoryless (complex-valued) AWGN channel model

$$y_k = x_k + n_k \quad (2.1)$$

for  $k \in \mathbb{Z}$ , where  $x_k \in \mathbb{C}$  denotes an information symbol,  $n_k$  is a realization of a zero-mean complex Gaussian random variable  $N_k$  with  $\mathbb{E}[N_k N_{k'}^*] = N_0 \delta[k - k']$  and power spectral density (PSD)  $N_0$ , and  $y_k \in \mathbb{C}$  is the channel output or simply the observation. The channel from  $x_k$  to  $y_k$  is characterized by the conditional PDF

$$f_{Y_k|X_k}(y_k|x_k) = \frac{1}{\pi N_0} \exp\left(-\frac{|y_k - x_k|^2}{N_0}\right). \quad (2.2)$$

The primary goal of this section is to motivate (2.1) in the context of fiber-optical communication systems. In particular, our target application is coherent, long-haul data transmission over SMFs. The validity of (2.1) in this case depends heavily on the assumed system parameters, e.g., the type of dispersion-compensation scheme that is being used.

## 2.2 The Nonlinear Schrödinger Equation

The starting point for fiber-optical channel modeling is the NLSE, which can be derived from Maxwell's equations under some assumptions that are appropriate for SMFs [29]. The NLSE is a partial differential equation that defines the input–output relationship for optical baseband signals<sup>1</sup> propagating through SMFs.

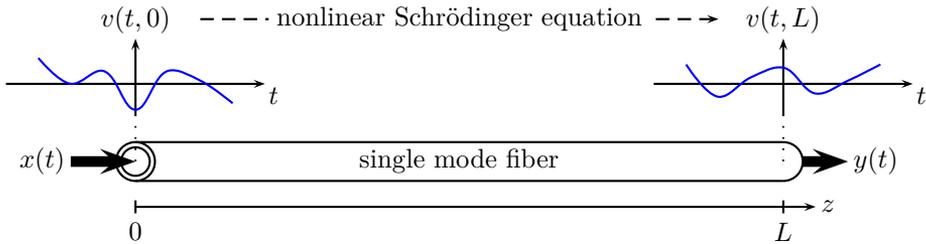
Let us first introduce a continuous-time parameter  $t \in \mathbb{R}$  and a distance parameter  $0 \leq z \leq L$  that denotes the propagation distance of the signal from the beginning of the fiber, where  $L$  is the total length of the fiber. The baseband signal of interest is a function of two parameters, denoted by  $v(t, z)$ . We define the input and output signals as  $x(t) \triangleq v(t, 0)$  and  $y(t) \triangleq v(t, L)$ , i.e.,  $x(t)$  is the signal launched into the fiber at  $z = 0$ , and  $y(t)$  is the signal received after propagating through an SMF of length  $L$ . This is conceptually illustrated in Fig. 2.1. Before we continue, we also define the instantaneous signal power  $P(t, z) \triangleq |v(t, z)|^2$  and the power profile  $P(z) \triangleq \lim_{T \rightarrow \infty} (\int_{-T}^T P(t, z) dt) / (2T)$ , where  $P = P(0)$  is the power of the input signal.

The NLSE accounts for signal attenuation, chromatic dispersion, and nonlinear effects in an SMF and can be written as

$$\frac{\partial v(t, z)}{\partial z} = -\frac{\alpha}{2} v(t, z) - j \frac{\beta_2}{2} \frac{\partial^2 v(t, z)}{\partial t^2} + j \gamma v(t, z) |v(t, z)|^2, \quad (2.3)$$

where  $\alpha$  is the attenuation coefficient,  $\beta_2$  is the chromatic dispersion coefficient, and  $\gamma$  is the nonlinear Kerr parameter. If we take into account only the first term on the

<sup>1</sup>Often called “slowly varying envelope” in the literature. The carrier frequency is assumed to be the equivalent of a 1550 nm light wave, corresponding to roughly 193.4 THz.



**Figure 2.1:** Conceptual representation of the signal evolution through an SMF. The NLSE describes the relationship between the input signal  $x(t) = v(t, 0)$  and the output signal  $y(t) = v(t, L)$ .

right-hand side of (2.3), one obtains  $v(t, z) = \exp(-\alpha z/2)v(t, 0)$  as a solution<sup>2</sup>, i.e., we immediately see that the signal amplitude in an SMF decays exponentially with the propagation distance. By defining a renormalized version of  $v(t, z)$  as  $u(t, z) \triangleq \exp(\alpha z/2)v(t, z)$  and substituting it into (2.3), one obtains an alternative and somewhat simpler version of the NLSE as [29, eq. (4)]

$$\frac{\partial u(t, z)}{\partial z} = -j\frac{\beta_2}{2}\frac{\partial^2 u(t, z)}{\partial t^2} + j\gamma e^{-\alpha z}u(t, z)|u(t, z)|^2. \quad (2.4)$$

In general, there are no closed-form solutions to the NLSE and one has to resort to numerical methods in order to obtain a solution. In the following, we briefly describe one of the most widely used numerical methods to solve (2.4), namely the split-step Fourier method (SSFM). Conceptually, we start by discretizing the spatial dimension and subdividing the entire fiber of length  $L$  into small segments of length  $\Delta$ , where  $M = L/\Delta \in \mathbb{N}$  is the total number of segments. For the  $i$ -th segment,  $1 \leq i \leq M$ , the input signal is denoted by  $u(t, (i-1)\Delta)$  and the corresponding output signal by  $u(t, i\Delta)$ . It is then assumed that an approximate solution to obtain  $u(t, i\Delta)$  based on  $u(t, (i-1)\Delta)$  is given by

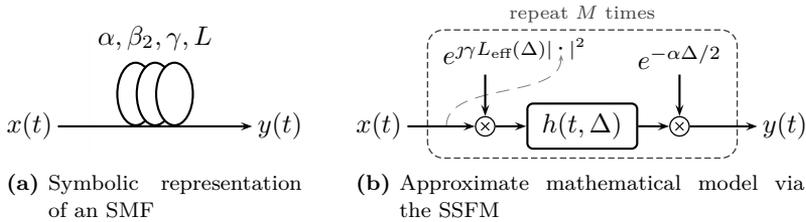
$$u(t, i\Delta) \approx h(t, \Delta) \otimes \left( u(t, (i-1)\Delta) e^{j\gamma L_{\text{eff}}(\Delta)|u(t, (i-1)\Delta)|^2} \right), \quad (2.5)$$

where  $h(t, z) = \exp(jt^2/(2\beta_2 z)) / \sqrt{j2\pi\beta_2 z}$  is the impulse response of a linear filter representing dispersive effects and

$$L_{\text{eff}}(z) \triangleq \int_0^z e^{-\alpha z'} dz' = \frac{1 - \exp(-\alpha z)}{\alpha} \quad (2.6)$$

is referred to as the effective nonlinear length with  $L_{\text{eff}}(z) \leq z$  and  $L_{\text{eff}}(z) \rightarrow z$  as  $\alpha \rightarrow 0$ . The reasoning behind (2.5) is that over a short segment of length  $\Delta \ll L$ , the linear (i.e., dispersive) and nonlinear effects act almost independently of one another.

<sup>2</sup>Recall that the solution of  $\partial f(z)/\partial z = cf(z)$  is given by  $f(z) = \exp(cz)f(0)$ .



**Figure 2.2:** Illustrations for an SMF. The notation  $|\cdot|^2$  in (b) stands for the instantaneous power of the signal that arrives at the corresponding multiplication block as indicated by the dashed, gray line.

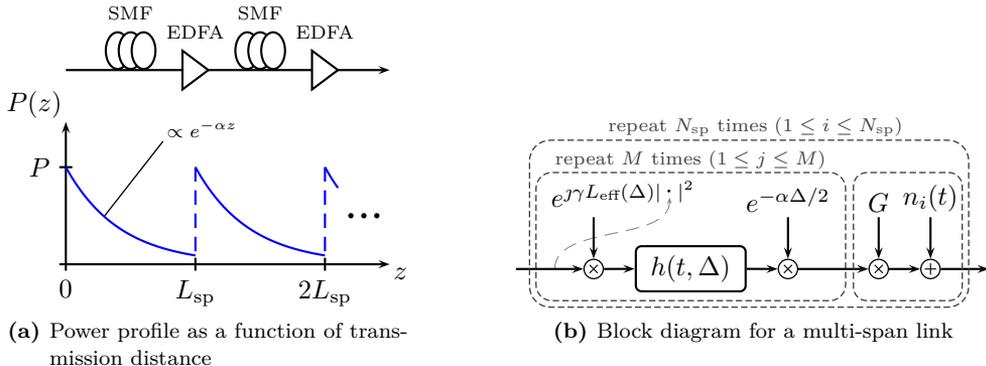
Using this assumption, an approximate solution for an entire SMF of length  $L$  is given by repeatedly applying (2.5), starting with the first segment where  $i = 1$ , i.e., with the input signal  $u(t, 0) = x(t)$ . The SSFM step in (2.5) is given in terms of the normalized signal  $u(t, z)$ . In order to incorporate the signal attenuation, the output signal  $u(t, i\Delta)$  is multiplied by  $\exp(-\alpha\Delta/2)$  to obtain  $v(t, i\Delta)$  after each step. Fig. 2.2 shows the resulting numerical method in terms of a block diagram. In the figure, the notation  $|\cdot|^2$  stands for the instantaneous power of the signal that arrives at the corresponding multiplication block (e.g.,  $|x(t)|^2$  in the first segment,  $|u(t, \Delta)\exp(-\alpha\Delta/2)|^2$  in the second, and so on). It has been shown that the above method converges to the true solution for  $\Delta \rightarrow 0$  [28, p. 42]. Practical guidelines on the choice of the segment size are developed in [30].

The name of the method originates from the fact that the nonlinear phase-shift operation and the linear filtering in Fig. 2.2(b) are commonly carried out in the time and frequency domain, respectively. Therefore, one forward and one inverse Fourier transform have to be performed per segment. In computer implementations, a sampled version of the baseband signal  $u(t, z)$  (or  $v(t, z)$ ) is considered which facilitates the application of the computationally efficient fast Fourier transform. Such an implementation is for example provided in [28, App. B].

## 2.3 Optical Amplification and Noise

The numerical value of the attenuation coefficient  $\alpha$  is typically between 0.2 and 0.4 dB/km. Assuming  $\alpha = 0.2$  dB/km and a transmission distance of  $L = 2000$  km, the input signal would be attenuated by 400 dB implying that  $y(t)$  is practically zero [2, Sec. IX-B]. It is therefore necessary to amplify the signal along the transmission path, which invariably introduces noise into the system.

We briefly discuss one common type of amplification, referred to as lumped amplification, in terms of its effect on the power profile of the signal and the type of noise that is introduced. Modeling the power profile is important due to the dependency of the nonlinear effect on the instantaneous signal power. Thus, one cannot simply ignore atten-



**Figure 2.3:** Illustrations for a fiber-optical communication link including a lumped amplification scheme and noise.

uation effects and make a link budget analysis as is common for linear channels. Details about the underlying physical aspects of optical amplification can be found in standard textbooks on optical data transmission, e.g., [31, Ch. 6]. It should, however, be pointed out that the optical amplifier noise is in fact the dominant source of noise in long-haul systems. This means that noise from other sources, e.g., thermal noise from electrical components, is negligible in comparison and can therefore be ignored [2, Sec. IX-A].

To account for amplification and noise, the NLSE (2.3) can be extended by inserting a gain profile  $g(z)$  and a complex-valued stochastic process  $w(t, z)$ , resulting in

$$\frac{\partial v(t, z)}{\partial z} = -\frac{\alpha - g(z)}{2}v(t, z) - j\frac{\beta_2}{2}\frac{\partial^2 v(t, z)}{\partial t^2} + j\gamma v(t, z)|v(t, z)|^2 + w(t, z). \quad (2.7)$$

Equation (2.7) is referred to as the stochastic nonlinear Schrödinger equation (sNLSE) [32]. We start by discussing the gain profile  $g(z)$  and its effect on the power profile of the signal  $v(t, z)$ , ignoring all other effects (including  $w(t, z)$ ). Signal amplification is applied periodically, in the sense that the entire transmission distance  $0 \leq z \leq L$  is split up into spans of length  $L_{\text{sp}}$  varying between 60 and 120 km, where  $N_{\text{sp}} = L/L_{\text{sp}} \in \mathbb{N}$  denotes the total number of spans. For lumped amplification, an optical amplifier, most often an erbium-doped fiber amplifier (EDFA) [2, Sec. IX-B], is inserted after each span, where the amplifier gain  $G$  matches the power loss of the signal in that span, i.e.,  $G = e^{\alpha L_{\text{sp}}}$ . In (2.7), this is accounted for by setting  $g(z) = \alpha L_{\text{sp}} \sum_{i=1}^{N_{\text{sp}}} \delta(z - iL_{\text{sp}})$ . The corresponding power profile is schematically illustrated in Fig. 2.3(a). It can be seen that the signal power decreases exponentially according to the loss coefficient  $\alpha$  and is periodically restored to the input power  $P$  after each span.

Next, we discuss the noise that is generated by optical amplification through a process called amplified spontaneous emission (ASE). Noise can be thought of as being added to the signal at discrete locations  $z_i \triangleq iL_{\text{sp}}$ ,  $1 \leq i \leq N_{\text{sp}}$ . Thus, if we think about  $z_i^-$  and  $z_i^+$  as the locations right before and after the amplifiers, we have  $v(t, z_i^+) = Gv(t, z_i^-) + n_i(t)$ ,

where  $n_i(t)$  is the additive noise originating from the  $i$ -th amplifier [33, p. 36]. It has been experimentally verified that ASE noise can be accurately modeled as circularly symmetric complex Gaussian [2, p. 667] and therefore it remains to specify the autocorrelation function of  $n_i(t)$ , where processes from different amplifiers are uncorrelated. The most common assumption is white Gaussian noise, i.e.,  $\mathbb{E}[N_i(t)N_j^*(t')] = N_\ell\delta(t-t')\delta[i-j]$ , where  $N_\ell$  denotes the noise PSD per amplifier. (The index  $\ell$  refers to the lumped amplification type.) We further set  $N_0 = N_{\text{sp}}N_\ell$ , which one might think of as the cumulative PSD at the end of the transmission link for  $N_{\text{sp}}$  amplifiers. Since temporally white noise has infinite instantaneous power, this assumption would, however, lead to infinite phase rotations due to the nonlinear effect. In reality, the noise power is of course finite, and the PSD of ASE noise is comparable to the gain spectrum of the amplifier. For an idealized EDFA that provides flat gain over a certain frequency range  $W_n$ , one would then replace  $\delta(t-t')$  with  $\delta_{W_n}(t-t')$  where  $\delta_{W_n}(x) = W_n\text{sinc}(W_n x)$  [33]. Further limitations of the optical bandwidth can occur due to the insertion of optical bandpass filters and/or reconfigurable optical add-drop multiplexer along the transmission line [2].

Based on the previous description, a block diagram of a continuous-time model for a multi-span transmission link with lumped amplification is depicted in Fig. 2.3(b). The model consists of the concatenation of the deterministic model for an SMF based on the SSFM (cf. Fig. 2.2(b)) with a multiplicative gain factor and additive noise representing the optical amplifier. For completeness, we also indicate how the additive noise terms  $n_i(t)$  can be related to  $w(t, z)$  in (2.7). Note that if we neglect all terms on the right-hand side of (2.7) except  $w(t, z)$ , we have  $\partial v(t, z)/\partial z = w(t, z)$  and integrating this equation leads to

$$v(t, z) = v(t, 0) + \int_0^z w(t, \xi) d\xi = v(t, 0) + n(t, z). \quad (2.8)$$

Here,  $n(t, z)$  represents the noise that is added to the signal up to a certain distance  $z$ . For lumped amplification, one may set  $w(t, z) = \sum_{i=1}^{N_{\text{sp}}} n_i(t)\delta(z - iL_{\text{sp}})$  [29, p. 84], so that  $n(t, z) = \sum_{i=1}^{\lfloor z/L_{\text{sp}} \rfloor} n_i(t)$  corresponds the addition of all  $n_i(t)$  up to distance  $z$  (the upper integral limit in (2.8) is interpreted as  $z^+$ ).

## 2.4 Polarization Multiplexing

In addition to the amplitude and phase (or, alternatively, the in-phase and quadrature component), data may also be encoded using the polarization of the light source. Systems where both polarizations of the light are used to transmit data are referred to as PM. For PM transmission, the sNLSE equation can be further extended by considering the vector signal  $\mathbf{v}(t, z) = (v_a(t, z), v_b(t, z))^T$ , where the indices indicate the two polarizations

**a** and **b**.<sup>3</sup> The resulting equation is referred to as the Manakov equation which includes amplifier noise, gain, and loss terms. It is given by [34, p. 8]

$$\frac{\partial \mathbf{v}(t, z)}{\partial z} = -\frac{\alpha - g(z)}{2} \mathbf{v}(t, z) - j \frac{\beta_2}{2} \frac{\partial^2 \mathbf{v}(t, z)}{\partial t^2} + j \gamma \mathbf{v}(t, z) \|\mathbf{v}(t, z)\|^2 + \mathbf{w}(t, z), \quad (2.9)$$

where  $\mathbf{w}(t, z) = (w_a(t, z), w_b(t, z))^T$  are two (independent) stochastic processes describing the ASE noise generated in both polarizations. The major difference between (2.9) and (2.7) is that (2.9) models the nonlinearity that is due to the sum of the instantaneous power in both polarizations  $\|\mathbf{v}(t, z)\|^2 = P_a(t, z)^2 + P_b(t, z)^2$ . We should mention that (2.7) ignores the fact that amplifier noise is always generated “in two polarizations”, i.e., even if we assume one of the two signals in  $\mathbf{v}(t, z)$  to be zero, technically the amplifier noise in that polarization still contributes via (2.9) through the fiber nonlinearity.

For simplicity, we ignore polarization-specific impairments. This includes for example polarization mode dispersion, which would cause different group velocities of the signals in polarization **a** and **b** caused by natural imperfections and asymmetries of the fiber cross-section area.

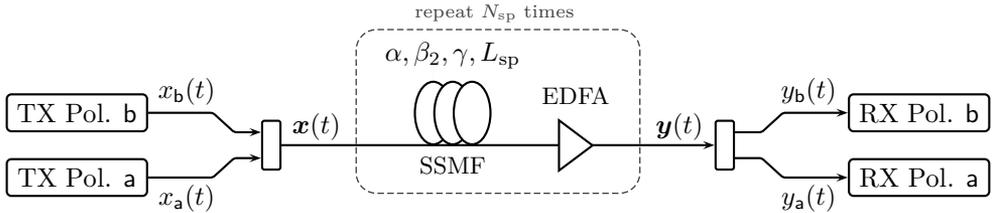
## 2.5 Linear Pulse Modulation and Linear Receiver

So far, we have discussed models for waveform channels. In order to arrive at a discrete-time channel model, we have to make some assumptions about the type of modulation that is used in the transmitter and the type of receiver structure. The statistics of the resulting discrete-time channel may depend heavily on these assumptions.

In Fig. 2.4, a generic block diagram for a PM transmission scheme is shown. We assume that the transmitters (TX) employ a linear pulse modulation according to  $x_a(t) = \sum_k x_{a,k} p(t - kT_s)$  for polarization **a** and similarly for polarization **b**, where  $T_s$  is the symbol period. The evolution of the PM signal is then described by the Manakov equation (2.9), where  $v_a(t, 0) = x_a(t)$  and  $v_b(t, 0) = x_b(t)$ . The received signal in each polarization is assumed to be processed according to a linear receiver. In particular, for polarization **a**, it is assumed that  $y_a(t) = v_a(t, L)$  is passed through an equalizer, a pulse-matched filter, and a sampler, in order to obtain  $y_{a,k'} = y_a(t) \otimes h(t, -L) \otimes p(-t)|_{t=k'T_s}$  and similarly for polarization **b**.

Characterizing the statistical relationship between the transmitted symbols and received samples is a challenging task due to the complicated interaction of the signal with itself, the noise, and the signal in the orthogonal polarization. Here, we focus on optical transmission links without any inline dispersion compensation, which are referred to as non-dispersion-managed or uncompensated transmission links. Recently, there has been a substantial amount of work on this type of transmission link with the goal to find such

<sup>3</sup>This nonstandard notation for the polarizations is an attempt to avoid confusion with the transmitted and received signals.



**Figure 2.4:** Block diagram of the PM transmission scheme considered in Papers A and B.

a statistical relationship [35–38].<sup>4</sup>

In [35], it is shown that the discrete-time channel for non-dispersion-managed links is well modeled by a circularly symmetric complex additive Gaussian channel including a complex scaling factor. In the derivation of the model, the assumption is that dispersive effects are dominant (i.e., the symbol rate  $1/T_s$  is high enough) and that the nonlinear effects are not too strong. The complex scaling accounts for a constant phase offset as well as the fact that part of the signal is converted into noise-like interference through the interaction between the dispersive and nonlinear effects. For simplicity, it is then assumed that this nonlinear noise is additive, Gaussian, and uncorrelated (both in time and across polarizations). A discrete-time channel model in polarization a is then given according to

$$y_a = \zeta x_{a,k} + n_{a,k} + \tilde{n}_{a,k}, \quad (2.10)$$

where  $\zeta \in \mathbb{C}$  is a complex scaling factor,  $n_{a,k}$  corresponds to the linear ASE noise with  $\mathbb{E}[N_{a,k}N_{a,k}^*] = N_0/T_s = P_{\text{ASE}}$ ,  $\tilde{n}_{a,k}$  accounts for nonlinear noise with  $\mathbb{E}[\tilde{N}_{a,k}\tilde{N}_{a,k}^*] = \eta P^3$ , and the same transmit power  $P$  is assumed for the signals in both polarizations. The parameter  $\eta$  (and hence the nonlinear noise variance) is a function of the link parameters and the symbol time, i.e.,  $\eta = f(\alpha, \beta_2, \gamma, L_{\text{sp}}, N_{\text{sp}}, T_s)$  [35, eq. (15)], and  $|\zeta|^2 = 1 - |\eta|P^2$ .

The main difference with respect to the “conventional” discrete-time additive Gaussian channel in (2.1) is that the signal-to-noise ratio (SNR) (defined as the ratio of the input power to the *additive* noise power) is not sufficient to characterize the operating point of the channel but rather one needs to consider the pair  $(P, P_{\text{ASE}})$  or, more practically relevant, the pair  $(P, L)$ . This parameter pair leads in turn to both a linear and a nonlinear noise variance based on which an effective SNR can be computed.

<sup>4</sup>This case is also of high practical relevance and according to [36], “the current consensus is that green-field installations, as well as major overhauling and refurbishing of existing links, should adopt uncompensated transmission.”

---

## Bit-Interleaved Coded Modulation

---

In this chapter, we provide a brief introduction on how to design systems that reliably transmit data at high spectral efficiencies. Spectrally-efficient communication can be achieved in practice by combining error-correcting codes with nonbinary signal constellations, which is commonly referred to as coded modulation (CM). We focus on BICM, which is a pragmatic approach to CM and often implemented in practice, due to its inherent simplicity and flexibility.

We start by outlining the main principles behind CM in Section 3.1. In Section 3.2, we explain the building blocks of a BICM system. We also cover the parallel independent channel model for BICM which is used for the bit mapper optimization problem studied in Papers A and B.

### 3.1 Introduction to Coded Modulation

Consider again the discrete memoryless AWGN channel in (2.1). The goal is to reliably transmit data at high spectral efficiencies over this channel. To do so, one can formally define an encoder  $\mathcal{E} : \{0, 1\}^d \rightarrow \mathcal{C}_c$ , which maps a vector of  $d$  information bits to a codeword in the code  $\mathcal{C}_c \subset \mathbb{C}^N$ . Each codeword is a complex vector of length  $N$  and its components serve as the input for  $N$  consecutive uses of the AWGN channel. Similarly, one can define a decoder  $\mathcal{D} : \mathbb{C}^N \rightarrow \{0, 1\}^d$ , which maps a vector of  $N$  channel observations back to a sequence of  $d$  estimated bits. Assuming equally likely information bits, the communication rate (measured in [bits/complex symbol]) of such a system is given by  $\kappa = \log_2(|\mathcal{C}_c|)/N = d/N$ . Notice that the communication rate of the discrete-time channel is intimately related to the spectral efficiency of the continuous-time channel (in [bits/s/Hz]) via the bandwidth of the pulse shape  $p(t)$  and the symbol time  $T_s$ . Shannon proved that all rates up to the channel capacity

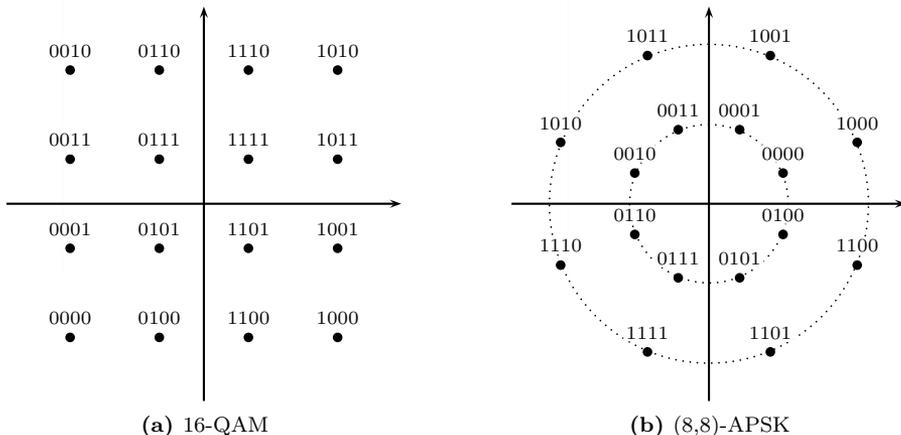
$$C = \log_2(1 + \text{SNR}) \quad (3.1)$$

are achievable, in the sense that there exists an encoder/decoder pair that can provide an arbitrarily small error probability as long as  $N \rightarrow \infty$  [39].

While Shannon's proof provides communication engineers with an invaluable benchmark, the problem of designing practical encoders and decoders that operate close to capacity and are implementable with reasonable complexity was not directly addressed by Shannon. In practical systems, the channel input  $x_k$  commonly does not take on arbitrary complex values, but is constrained to a discrete signal constellation  $\mathcal{X} \subset \mathbb{C}$ . Given this premise, it is useful to introduce a soft dividing line between two different operating regimes for this channel. This dividing line is at  $\kappa = 2$ , where  $\kappa \leq 2$  is referred to as the power-limited regime and  $\kappa > 2$  as the bandwidth-limited regime [40]. Roughly speaking, in the power-limited regime, it is sufficient to consider a binary modulation independently in the real and imaginary part (e.g., Gray-labeled quadrature phase-shift keying (QPSK) according to  $\mathcal{X} = \{1 + j, 1 - j, -1 + j, -1 - j\}$  and scaled by  $\sqrt{P/2}$ ), in combination with binary error-correcting codes in order to operate close to the capacity. On the other hand, spectrally-efficient communication requires the use of signal constellations with cardinality larger than 4, which are referred to as nonbinary or higher-order<sup>1</sup> constellations. By invoking the capacity formula, it follows directly that operating at high spectral efficiencies (where  $\kappa > 2$ ) requires the signal power to be at least three times the noise power. In other words, spectrally-efficient communication requires a reasonably high SNR.

Devising practical encoder/decoder pairs where  $x_k$  is constrained to a higher-order signal constellation is commonly referred to as CM design. There exist several different approaches, for example trellis coded modulation [41], CM with nonbinary codes [42],

<sup>1</sup>One may also classify complex constellations with 4 points as "higher-order", as long as they cannot be viewed as two independent binary modulations per real and complex dimension.



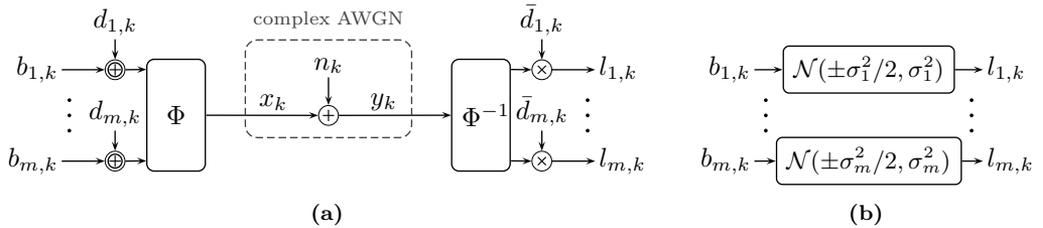
**Figure 3.1:** Two examples of higher-order signal constellations with 16 points.

multilevel coded modulation [43], or BICM [44]. Our focus here is on BICM in combination with (binary) LDPC codes, which is one of the most popular capacity-approaching coding schemes for achieving high spectral efficiency, due to its simplicity and flexibility [45]. BICM is employed as the *de facto* standard in many wireless communication standards and has also been studied by many authors for fiber-optical communication systems, see, e.g., [46] or [47] and references therein.

## 3.2 BICM System Model

In the following, the transmitted symbols  $x_k$  in each time instant  $k$  are assumed to take on values from a discrete signal constellation  $\mathcal{X} \subset \mathbb{C}$  with  $|\mathcal{X}|$  points, where  $|\mathcal{X}|$  is a power of two. Furthermore, each point in the constellation is assumed to be labeled with a unique binary string of length  $m = \log_2 |\mathcal{X}|$ , where  $b_i(x)$ ,  $1 \leq i \leq m$ , denotes the  $i$ -th bit in the binary string assigned to  $x \in \mathcal{X}$  (counting from left to right). Two examples of signal constellations with  $|\mathcal{X}| = 16$  points are shown in Fig. 3.1 and referred to as 16-QAM and (8, 8)-APSK.

We now describe the main components of a BICM system. First, consider the block diagram shown in Fig. 3.2(a), where the modulo 2 addition of  $d_{i,k} \in \{0, 1\}$  and multiplication by  $\tilde{d}_{i,k} = (-1)^{d_{i,k}}$  are explained further below and can be ignored for now. At each time instant, the modulator  $\Phi$  takes  $m$  bits  $b_{i,k}$ ,  $1 \leq i \leq m$ , and maps them to one of the constellation points according to the binary labeling of the signal constellation. At the receiver, the demodulator  $\Phi^{-1}$  computes soft reliability information about the



**Figure 3.2:** (a) The modulator  $\Phi$ , demodulator  $\Phi^{-1}$ , and channel symmetrization technique. (b) A helpful approximate channel model via parallel symmetric Gaussian LLR channels.

transmitted bits in the form of the log-likelihood ratios (LLRs)

$$l_{i,k} \triangleq \log \frac{f_{Y_k|B_{i,k}}(y_k|0)}{f_{Y_k|B_{i,k}}(y_k|1)} = \log \frac{\sum_{x \in \mathcal{X}_{i,0}} f_{Y_k|X_k}(y_k|x)}{\sum_{x \in \mathcal{X}_{i,1}} f_{Y_k|X_k}(y_k|x)}, \quad (3.2)$$

where  $\mathcal{X}_{i,u} \triangleq \{x \in \mathcal{X} : b_i(a) = u\}$  is the subconstellation where all points have the bit  $u$  at the  $i$ -th position of their binary label. The LLR is a function of the observation and, since the observation is a random variable, the LLR is also a random variable.

One way to interpret the setup depicted in Fig. 3.2(a) is as follows. The concatenation of the modulator  $\Phi$ , the AWGN channel, and demodulator  $\Phi^{-1}$  establishes a binary interface for the complex-valued AWGN channel. It is useful to imagine transmitting data over a set of  $m$  parallel binary-input continuous-output channels, or simply bit channels, where one may view the conditional distribution of the LLR  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$ ,  $1 \leq i \leq m$ , as a bit channel. In the following, a bit channel  $f_{L|B}(l|b)$  is called symmetric if  $f_{L|B}(l|0) = f_{L|B}(-l|1)$  and referred to as an LLR channel if  $f_{L|B}(l|0)e^l = f_{L|B}(l|1)$ . The terminology is used to emphasize that, if the second condition is fulfilled, the output of the channel corresponds to a “true” LLR. This is important because, in practice, low-complexity approximations of (3.2) are often considered, and the resulting bit channel in that case is not necessarily an LLR channel [48, Ch. 5]. While  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  is an LLR channel, the channel is not necessarily symmetric in general.<sup>2</sup> Symmetry can be enforced by adding modulo 2 independent and identically distributed bits  $d_{i,k}$  to the bits  $b_{i,k}$  [49]. After the demodulator, the corresponding LLR is multiplied by  $\bar{d}_{i,k} = (-1)^{d_{i,k}}$ , which implies that the bits  $d_{i,k}$  are known to both the transmitter and receiver. The resulting bit channel  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  can be shown to be symmetric [49].

We proceed by quantifying the quality of the  $m$  bit channels, where we rely on the mutual information (MI) as a measure of quality. The MI between the output of a

<sup>2</sup>The symmetry condition will become important when discussing DE and LDPC codes, where one relies on the all-zero codeword assumption.

symmetric LLR channel  $f_{L|B}(l|b)$  and uniform input bits is given by

$$I(L; B) = \mathbb{E} \left[ \log_2 \left( \frac{f_{L|B}(L|B)}{f_L(L)} \right) \right] \quad (3.3)$$

$$= 1 - \mathbb{E} \left[ \log_2 \left( \frac{f_{L|B}(L|B) + f_{L|B}(L|1-B)}{f_{L|B}(L|B)} \right) \right] \quad (3.4)$$

$$= 1 - \mathbb{E} \left[ \log_2 \left( 1 + \frac{f_{L|B}(L|1-B)}{f_{L|B}(L|B)} \right) \right] \quad (3.5)$$

$$= 1 - \mathbb{E} \left[ \log_2 (1 + \exp((-1)^{1-B}L)) \right] \quad (3.6)$$

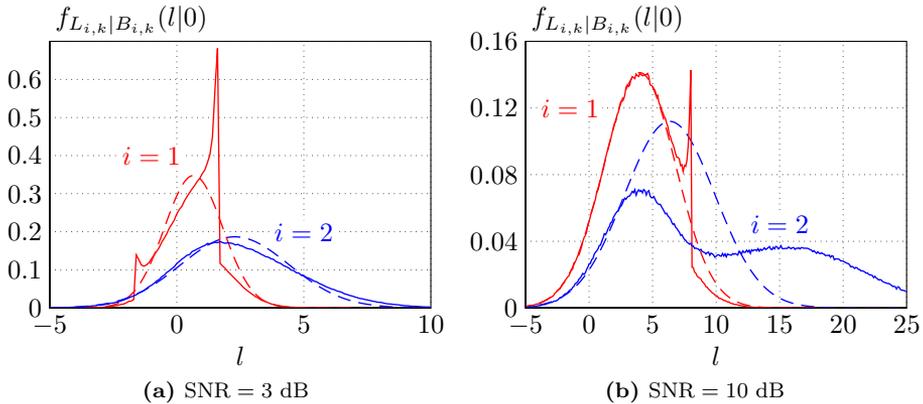
$$= 1 - \int_{-\infty}^{+\infty} f_{L|B}(l|0) \log_2(1 + \exp(-l)) dl. \quad (3.7)$$

Writing the MI in the form (3.7) can be useful in order to compute the MI with the help of Monte Carlo integration.

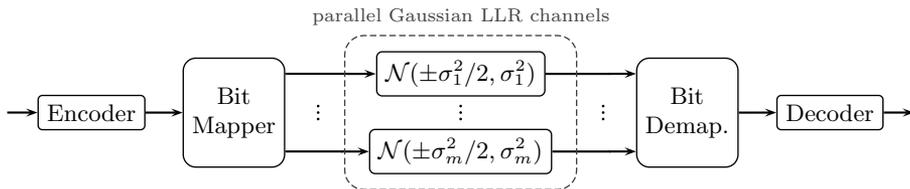
It turns out that, while the channel quality of the bit channels can be determined quite efficiently, it is very difficult to find exact analytical expressions for the actual densities  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$ . A common approach in the analysis of BICM is to make the simplifying assumption that the densities  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  are Gaussian. An LLR channel with a Gaussian density is particularly simple, because it can be parametrized by a single parameter. More precisely, we refer to a bit channel  $f_{L|B}(l|b)$  as a symmetric Gaussian LLR channel with parameter  $\sigma^2$  if  $L \sim \mathcal{N}(\sigma^2/2, \sigma^2)$  conditioned on  $B = 0$  and  $L \sim \mathcal{N}(-\sigma^2/2, \sigma^2)$  conditioned on  $B = 1$ , where  $\mathcal{N}(\mu, \sigma^2)$  denotes the Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ . The MI between the output of a symmetric Gaussian LLR channel and uniform input bits is denoted by  $J(\sigma)$ . Under the Gaussian assumption, a helpful approximation of the setup in Fig. 3.2(a) is shown in Fig. 3.2(b), where transmission takes place over  $m$  parallel symmetric Gaussian LLR channels with different parameters  $\sigma_i^2$ . In order to find a correspondence between the LLR channels  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  and the parameters  $\sigma_i^2$ , one may match the MI according to  $J(\sigma_i) = I_i(\text{SNR}) \Leftrightarrow \sigma_i^2 = J^{-1}(I_i(\text{SNR}))^2$ , where  $I_i(\text{SNR}) = I(B_{i,k}; L_{i,k})$  is independent of  $k$ .

While the parallel Gaussian model can be quite useful, one should, however, be aware of the inaccuracies of this simplified model. In particular, the bit channels are not independent as suggested in Fig. 3.2(b) and the true distribution of the LLRs is not Gaussian. To illustrate the latter inaccuracy, in Fig. 3.3, we compare the actual densities with the approximated Gaussian densities for two different SNRs for the first two bit positions of the 16-QAM constellation shown in Fig. 3.1(a).<sup>3</sup> The densities  $f_{L_{i,k}|B_{i,k}}(l|0)$  are estimated via histograms and shown by the solid lines, whereas the Gaussian densities are shown by the dashed lines. It can be seen that the actual densities are clearly non-Gaussian and the accuracy of the Gaussian approximation therefore depends on the application scenario. For the application in Papers A and B (i.e., predicting the iterative

<sup>3</sup>The third and fourth bit positions lead to identical distributions, due to the fact that 16-QAM with the shown labeling can be seen as a product constellation of two one-dimensional constellations.



**Figure 3.3:** Comparison of the true LLR channels (including channel symmetrization) with the symmetric Gaussian LLR channels that have the same MI.



**Figure 3.4:** A useful approximate system model for BICM systems.

performance behavior of LDPC codes), the approximation turns out to be quite accurate and at the same time allows for a major simplification of the analysis, thereby justifying its use.

Consider now the case where we employ a single binary code  $\mathcal{C} \subset \{0, 1\}^n$  of length  $n$ , and each codeword is transmitted using  $N = n/m$  symbols  $x_k$ . The allocation of the coded bits to the modulator (i.e., the different bit channels in Fig. 3.2(b)) is determined by a bit mapper as shown in Fig. 3.4. In Papers A and B, our goal is to find good bit mappers for a given code and signal constellation.

As a side note, we remark that the term “bit interleaver” is also commonly used instead of “bit mapper”. In fact, the modulator  $\Phi$  is sometimes referred to as the (symbol) mapper (and the demodulator  $\Phi^{-1}$  as the demapper), which the reader should be aware of in order to avoid confusion. However, the terms “bit mapper”, “bit mapping”, or “mapping” seem to be preferred in the literature when the allocation of the coded bits to the constellation symbols  $\Phi$  is explicitly studied or optimized, see, e.g., [50, 51]. Moreover, outside the context of BICM, the terms “mapping device” or “channel mapper” are used when studying parallel channels in combination with binary codes, e.g., in [52, 53].

---

## Low-Density Parity-Check Codes

---

LDPC codes were proposed by Gallager in his Ph.D. thesis [15]. They were conceived as practically decodable codes, able to “utilize the long block lengths necessary for low error probability without requiring excessive equipment or computation” [54].

In this chapter, we review some basic concepts behind LDPC codes and iterative decoding, focusing on protograph-based codes. In Section 4.1, we give a formal definition of an LDPC code. In Section 4.2, we review BP decoding which is based on message passing. The protograph-based construction of LDPC codes is explained in 4.3. The basic idea behind the asymptotic analysis of LDPC codes via DE is outlined in Section 4.4. Finally, in Section 4.5 we briefly cover spatially-coupled LDPC codes, which are one of the code classes considered for the problem statement addressed in Papers A and B.

## 4.1 Introduction

A binary LDPC code  $\mathcal{C}$  of length  $n$  is defined as the null space of a sparse parity-check matrix  $\mathbf{H} = [h_{i,j}] \in \{0, 1\}^{c \times n}$ , i.e.,

$$\mathcal{C} = \{\mathbf{c} \in \{0, 1\}^n : \mathbf{H}\mathbf{c}^\top = \mathbf{0}\}, \quad (4.1)$$

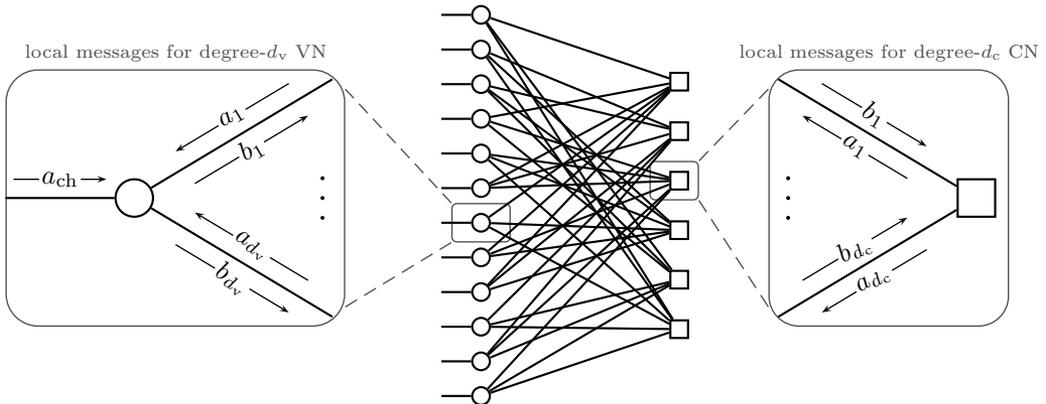
where  $n > c$  and operations (i.e., additions and multiplications) are over the binary field. Assuming that  $\mathbf{H}$  has full rank  $c$ , one can invoke the fundamental theorem of linear algebra to infer that the code has  $|\mathcal{C}| = 2^d$  codewords, where  $d = n - c$  is the dimension of the code. The code rate is defined as  $R = d/n = 1 - c/n$ .

The definition in (4.1) does indeed apply to an arbitrary binary linear code with a given parity-check matrix  $\mathbf{H}$ . It is of course up to interpretation when exactly the matrix  $\mathbf{H}$  should be classified as sparse (and, hence, the resulting code should be classified as an LDPC code). As an example, consider the case where  $\mathbf{H}$  is such that each row contains exactly  $d_c$  ones and each column contains exactly  $d_v$  ones. Choosing  $n$  and  $c$  large compared to  $d_c$  and  $d_v$  then leads to a sparse matrix  $\mathbf{H}$ . The code defined by such a matrix  $\mathbf{H}$  is referred to as a *regular* LDPC code.

## 4.2 Iterative Belief Propagation Decoding

Consider the scenario where each bit in the codeword of an LDPC code is transmitted over an LLR channel  $f_{L|B}(\cdot|\cdot)$  (recall the definition of an LLR channel in Section 3.2). The goal of the decoder is to recover the transmitted codeword based on the observation from the channel, which consists of  $n$  LLRs. These LLRs can be interpreted as the initial belief about each coded bit. The decoding is based on a graphical representation of the code. In particular, the parity-check matrix of an LDPC code can be represented by using a bipartite Tanner graph consisting of  $n$  variable nodes (VNs) and  $c$  constraint nodes (CNs), where the  $i$ -th CN is connected to the  $j$ -th VN if and only if  $h_{i,j} = 1$ . During the decoding process, the decoder tries to iteratively improve the accuracy of the initial belief by exchanging messages in the form of extrinsic LLRs between VNs and CNs along the edges of the Tanner graph.

For an excellent and comprehensive description of BP decoding, we refer the reader to [12, Ch. 5.3]. Here, we only briefly review the basic steps of the decoding algorithm. We use the following convention. Messages arriving at VNs are denoted by  $a$  and messages emanating from VNs are denoted by  $b$ . For CNs, it is the other way around, i.e., arriving messages are denoted by  $b$ , while emanating messages by  $a$ . In an attempt to avoid cluttered notation, only one index is appended to  $a$  or  $b$  in order to *locally* distinguish between messages along different edges for the same node. The corresponding picture we have in mind is illustrated in Fig. 4.1. By locally we mean that, for example, the message  $b_1$  emanating from the magnified VN does not correspond to the message  $b_1$  arriving at the magnified CN. (In fact, from the way the figure is drawn, the message  $b_1$  arriving at



**Figure 4.1:** Illustration of the messages involved in the iterative BP decoding algorithm.

the magnified CN would emanate from the fourth VN, counting from the top.)

Consider now an arbitrary VN of degree  $d_v$ , where the degree of a VN corresponds to the number of CNs that are connected to it. There are  $d_v + 1$  messages arriving at this VN, where  $a_1, \dots, a_{d_v}$  are messages from CNs and  $a_{ch}$  corresponds to the channel LLR. The  $d_v$  outgoing messages  $b_1, \dots, b_{d_v}$  are computed according to

$$b_i = \sum_{\sim i} a_j + a_{ch}, \quad (4.2)$$

where the summation is over the index set  $j \in \{1, \dots, d_v\}$  *excluding* the index  $i$ . Similarly, if we consider an arbitrary CN of degree  $d_c$ , there are  $d_c$  incoming messages  $b_1, \dots, b_{d_c}$  and the outgoing messages are computed according to

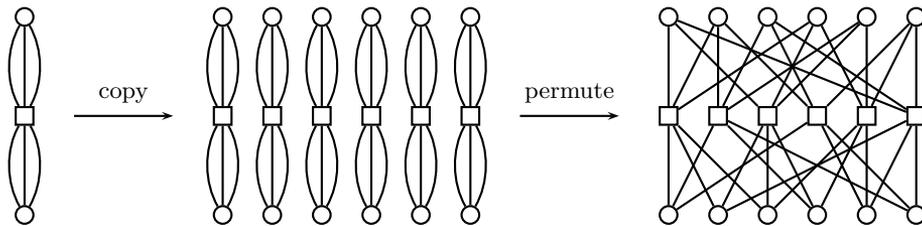
$$a_i = 2 \tanh^{-1} \left( \prod_{\sim i} \tanh(b_j/2) \right), \quad (4.3)$$

where the product is over the index set  $j \in \{1, \dots, d_c\}$  *excluding* the index  $i$ . Since the CN operation (4.3) is central to the analysis of LDPC codes under iterative decoding, it is very common to rewrite it in terms of the binary boxplus operator defined by

$$b_1 \boxplus b_2 = 2 \tanh^{-1} (\tanh(b_1/2) \tanh(b_2/2)). \quad (4.4)$$

The box-addition of an arbitrary number of terms is evaluated by recursively applying (4.4), e.g.,  $b_1 \boxplus b_2 \boxplus b_3 = (b_1 \boxplus b_2) \boxplus b_3$ . With this convention, one can write the CN operation more concisely as

$$a_i = \boxplus_{\sim i} b_j. \quad (4.5)$$



**Figure 4.2:** Illustration of the protograph lifting procedure for  $\mathbf{P} = (3, 3)$  and  $M = 6$ .

The decoding process can now be described as follows. Set  $a_{\text{ch}}$  for all VNs to the corresponding channel LLR and set all other messages to 0. (As an example, in a BICM system, the channel LLRs are computed according to (3.2).) Then, repeat the following two steps. First, compute outgoing messages for all VNs according to (4.2). After that, compute the outgoing messages for all CNs according to (4.3). Stop if either a maximum number of iterations has been reached or the proper combination of the hard decisions on the messages

$$\sum_{j=1}^{d_v} a_j + a_{\text{ch}} \quad (4.6)$$

for all VNs forms a valid codeword.

### 4.3 Code Construction via Protographs

There exist different methods to construct “good” LDPC codes, i.e., good matrices  $\mathbf{H}$ , and one popular method is by using protographs [20]. A protograph is a “small” bipartite graph defined by an adjacency matrix  $\mathbf{P} = [p_{i,j}] \in \mathbb{N}_0^{c' \times n'}$ , called the base matrix. Given  $\mathbf{P}$ , a parity-check matrix  $\mathbf{H}$  is obtained by replacing each entry  $p_{i,j}$  in  $\mathbf{P}$  with a random binary  $M$ -by- $M$  matrix which contains  $p_{i,j}$  ones in each row and column. This procedure is called lifting and  $M \geq \max_{i,j} p_{i,j}$  is the so-called lifting factor. Graphically, it amounts to copying the protograph  $M$  times and subsequently permuting edges in order to obtain the Tanner graph. Parallel edges, i.e., for  $p_{i,j} > 1$ , are permitted in the protograph and are resolved in the lifting procedure. The design rate of the code is given by  $R = 1 - c/n = 1 - c'/n'$ , where  $c = c'M$  and  $n = n'M$ . As an example, the lifting procedure for  $\mathbf{P} = (3, 3)$  and  $M = 6$  is illustrated in Fig. 4.2.

Designing codes via protographs has several practical advantages, e.g., a quasi-cyclic code is easily obtained by constraining the  $M$ -by- $M$  matrices to have a circulant structure. This in turn allows for hardware-efficient implementation [12, p. 263] suitable for high-speed optical communications [6]. Moreover, codes of different lengths can be obtained simply by adjusting the lifting factor.

## 4.4 Density Evolution

DE is a powerful tool to analyze the iterative decoding behavior and performance of LDPC codes in the limit of infinite block length [24]. DE mimics the decoding process under a cycle-free graph assumption by tracking how the densities of the messages evolve with iterations. DE is commonly used to find so-called decoding thresholds, which can be interpreted as the capacity for LDPC codes under BP decoding. Similar to the channel capacity, the threshold divides the channel quality parameter range (for example the parameter  $\sigma^2$  of a symmetric Gaussian LLR channel) into a region where reliable decoding is possible and where it is not.

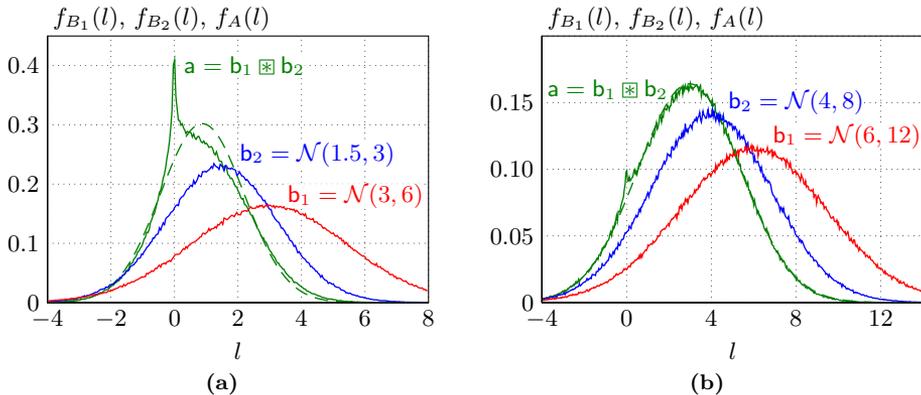
The main steps in the DE algorithm can be understood by considering the update equations (4.2) and (4.3) for the VNs and CNs, respectively. If we assume that the involved incoming messages are random variables, then they have a certain probability distribution or density. For example,  $a_{\text{ch}}$  is distributed according to the LLR channel. The main question is, how can we obtain the densities of the outgoing messages? For the VN update, the answer turns out to be a simple convolution. In particular, for two independent random variables  $A$  and  $B$  with distributions  $f_A(a)$  and  $f_B(b)$ , their sum  $C = A + B$  is distributed according to  $f_C(c) = f_A(a) \otimes f_B(b)$ , where  $\otimes$  denotes convolution. It is convenient to introduce the short notation  $\mathbf{a} \otimes \mathbf{b}$ , where  $\mathbf{a}$  and  $\mathbf{b}$  are placeholders for the densities of the random variables  $A$  and  $B$  [11]. Given the densities of the incoming messages, the densities of the outgoing messages can then be computed according to

$$\mathbf{b}_i = \underset{\sim i}{\star} \mathbf{a}_j \otimes \mathbf{a}_{\text{ch}}. \quad (4.7)$$

For the CN update, it is somewhat more challenging to obtain the densities of the outgoing messages. The most straightforward approach is by using Monte Carlo techniques and histograms. Consider the case where two messages  $b_1$  and  $b_2$  with densities  $\mathbf{b}_1$  and  $\mathbf{b}_2$  are processed according to the boxplus operation  $a = b_1 \boxplus b_2$ . In order to obtain the density  $\mathbf{a}$ , one can simply generate many independent realizations of the random variables  $B_1$  and  $B_2$ , perform the boxplus operation, and collect the resulting samples. These samples can be seen as a particle representation of the density  $\mathbf{a}$ . This method is illustrated in Fig. 4.3, where it is shown how two consistent Gaussian densities “evolve” under the boxplus operation. A density  $\mathbf{a}$  is called a consistent Gaussian density<sup>1</sup> with parameter  $\sigma^2$  if  $A \sim \mathcal{N}(\sigma^2/2, \sigma^2)$ . As a short notation, one may introduce the operator  $\mathbf{a} = \mathbf{b}_1 \boxtimes \mathbf{b}_2$ , which is referred to as box-convolution [11]. In practice, the box-convolution of two densities can be implemented by using a look-up table approach [55]. Similar to (4.7), the densities of the outgoing CN messages can then be computed according to

$$\mathbf{a}_i = \underset{\sim i}{\boxtimes} \mathbf{b}_j. \quad (4.8)$$

<sup>1</sup>Note that the conditional distribution  $f_{L|B}(l|0)$  of a symmetric Gaussian LLR channel corresponds to a consistent Gaussian density.



**Figure 4.3:** Illustration of the box-convolution of two consistent Gaussian densities. The green dashed line corresponds to the consistent Gaussian approximation obtained via EXIT functions.

For protograph-based codes, DE can be used to analyze the iterative decoding behavior by tracking one density for each edge in the protograph. This asserts that the messages exchanged during the decoding process over edges belonging to the same edge-type (defined by one protograph edge) have the same density. Assume that the transmission takes place over a symmetric LLR channel with a fixed channel quality. Due to the channel symmetry, one may assume the transmission of the all-zero codeword [12, p. 389]. The iterative decoding behavior can be predicted via DE as follows. Set  $\mathbf{a}_{\text{ch}}$  for all VNs in the protograph to  $f_{L|B}(l|0)$  and set all other densities to  $\delta(l)$ . Then, repeat the following two steps. First, calculate the outgoing message densities for all VNs in the protograph according to (4.7). After that, calculate the outgoing message densities for all CNs in the protograph according to (4.8). Stop if the error probability associated with the density

$$\bigotimes_{j=1}^{d_v} \mathbf{a}_j \boxtimes \mathbf{a}_{\text{ch}} \quad (4.9)$$

for each VN is below a certain target bit error probability (successful decoding), where the error probability associated with a density  $\mathbf{a}$  is given by

$$p_e(\mathbf{a}) = \int_{-\infty}^0 f_A(a) da, \quad (4.10)$$

or a maximum number of iterations is reached (decoding failure). In order to find the decoding threshold, we start from a channel quality where the decoding is predicted to be successful. The above procedure is then repeatedly applied for decreasing channel quality until the decoding fails.

### Approximate Density Evolution via EXIT Functions

Tracking the full densities (or quantized densities in practice) is computationally demanding and extrinsic information transfer (EXIT) functions are usually considered to be a good compromise between computational efficiency and accuracy [56]. Let us assume that the density  $\mathbf{a}$  fulfills the condition  $f_A(a)e^a = f_A(-a)$ . Then, the density can be associated with the MI measure

$$I(\mathbf{a}) = 1 - \int_{-\infty}^{\infty} f_A(a) \log_2(1 + e^{-a}) da. \quad (4.11)$$

Now, instead of tracking the evolution of densities, one may track the evolution of the MI measure associated with the densities (which is just a scalar value for each density). Let us assert that, under the VN operation, this measure evolves approximately according to

$$I(\mathbf{b}_i) \approx \tilde{J} \left( \sum_{\sim i} \tilde{J}^{-1}(I(\mathbf{a}_j)) + \tilde{J}^{-1}(I(\mathbf{a}_{\text{ch}})) \right), \quad (4.12)$$

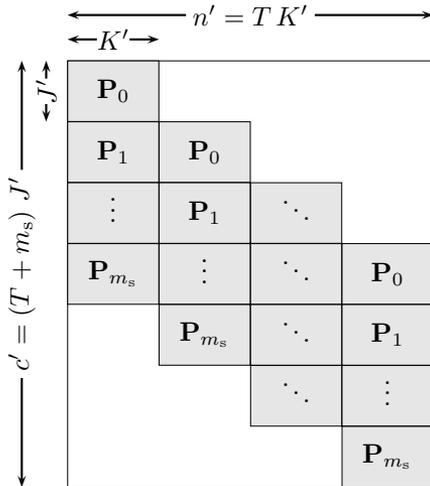
whereas, under the CN operation it evolves approximately according to

$$I(\mathbf{a}_i) \approx 1 - \tilde{J} \left( \sum_{\sim i} \tilde{J}^{-1}(1 - I(\mathbf{b}_j)) \right), \quad (4.13)$$

where  $\tilde{J}(x) = J(\sqrt{x})$ . These two equations can be motivated as follows. Eq. (4.12) is exact under the assumption that all incoming densities  $\mathbf{a}_1, \dots, \mathbf{a}_{d_v}$ , and  $\mathbf{a}_{\text{ch}}$  are consistent Gaussian densities. To see this, note that the convolution of two consistent Gaussian densities with parameters  $\sigma_1^2$  and  $\sigma_2^2$  is another consistent Gaussian density with parameter  $(\sigma_1^2 + \sigma_2^2)/2$ . Furthermore, if  $\mathbf{a}$  is a consistent Gaussian density with parameter  $\sigma^2$ , the operation  $\tilde{J}^{-1}(I(\mathbf{a}))$  simply returns  $\sigma^2$ . Without going into the details, (4.13) can be heuristically motivated by a duality property that holds for the binary erasure channel (BEC) [12, p. 415]. It is important to point out that (4.13) it is not exact, even if all incoming densities are consistent Gaussians, but it turns out to be surprisingly accurate nonetheless. For example, the green dashed lines in Fig. 4.3 have been obtained using (4.13), where the resulting MI measure is plotted in the form of the associated consistent Gaussian density.

## 4.5 Spatially-Coupled LDPC Codes

Spatial coupling of regular LDPC codes has emerged as a powerful technique to construct capacity-achieving codes for a large class of channels using iterative BP decoding [14, 57]. The main idea is to make several copies of the Tanner graph that defines the regular code, arrange the copies next to each other, and then interconnect neighboring graphs in a particular way. The key to the outstanding performance of codes constructed in such



**Figure 4.4:** Illustration of the base matrix  $\mathbf{P}_{[T]}$  of a  $(J, K)$  regular, protograph-based SC-LDPC code.

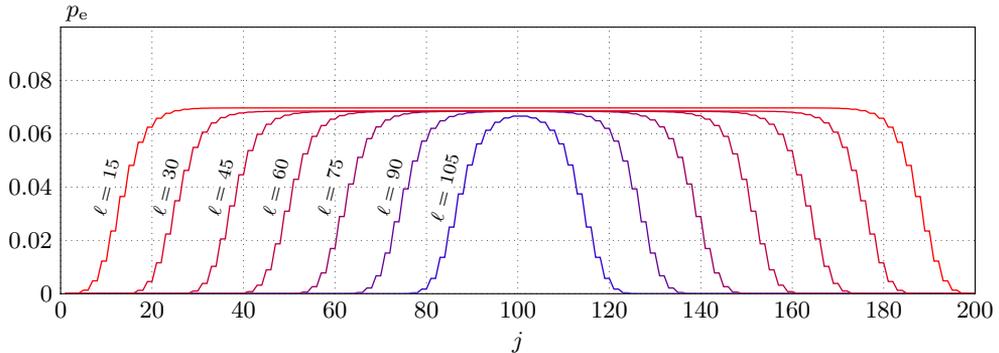
a way is a boundary effect due to slight irregularities at the two ends of the resulting Tanner graph.

In general, spatially-coupled LDPC codes have parity-check matrices with a band-diagonal structure, see, e.g., [57] for a formal definition. Here, we briefly introduce their construction via protographs [58], [59, Sec. II-B]. The base matrix  $\mathbf{P}_{[T]}$  of a  $(J, K)$  regular, protograph-based spatially-coupled LDPC code with termination length  $T$  can be constructed by specifying matrices  $\mathbf{P}_i$ ,  $0 \leq i \leq m_s$  of dimension  $J'$  by  $K'$ , where  $m_s$  is referred to as the memory. The matrices are such that  $\mathbf{P} = \sum_i \mathbf{P}_i$  has column weight  $J$  and row weight  $K$  for all columns and rows, respectively. Given  $T$  and the matrices  $\mathbf{P}_i$ , the base matrix  $\mathbf{P}_{[T]}$  is constructed as shown in Fig. 4.4. From the dimensions of  $\mathbf{P}_{[T]}$  one can infer a design rate of  $R(T) = 1 - (T + m_s)J'/(TK')$ . As  $T$  grows large, the rate approaches  $R(\infty) = 1 - J'/K'$ .

Before continuing, it is insightful to recall the following statement from [60], where the design of irregular LDPC codes is studied. (VNs are referred to as message nodes and CNs are referred to as check nodes.)

“[...] we offer some intuition as to why irregular graphs prove useful. [...] Message nodes with high degree tend to their correct value quickly. These nodes then provide good information to the check nodes, which subsequently provide better information to lower degree message nodes. Irregular graph constructions thus *lead to a wave effect*, where high degree message nodes tend to get corrected first, and then message nodes with slightly smaller degree, and so on down the line.” [emphasis added]

For spatially-coupled LDPC codes, one can give a similar heuristic explanation for their



**Figure 4.5:** Illustration of the wave-like decoding behavior of spatially-coupled LDPC codes.

outstanding performance as follows (see [57] for a detailed explanation). By inspecting the structure of the base matrix in Fig. 4.4, one may verify that the CN degrees corresponding to the first and last couple of rows are lower than the CN degrees corresponding to the rows in between. The lower degree CNs lead to a locally better decoding capability which helps decoding neighboring VNs. This local boundary effect turns out to initiate a wave-like behavior and can have a global effect on the decoding capability of the entire code with increasing number of decoding iterations. To illustrate this behavior, in Fig. 4.5, we show the predicted bit error rates  $p_e$  via (approximate) DE for the coded bits corresponding to the  $j$ -th column of the spatially-coupled LDPC protograph  $\mathbf{P}_{[T]}$  with component matrices  $\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}_3 = (1, 1)$  and  $T = 100$ . We assume transmission over a symmetric Gaussian LLR channel with parameter  $\sigma^2 = 4$ . In the figure,  $\ell$  denotes the iteration number. It can be observed that the error probability of the VNs at the two ends of the graph converges to zero after 15 iterations. Due to the spatial coupling, this boundary effect propagates inwards all the way to the center of the protograph in a wave-like fashion.

An important reason for the tremendous interest in spatially-coupled LDPC codes is their universality. While irregular LDPC codes have been optimized for various communication channels, the degree distribution pairs that achieve the best performance usually vary from channel to channel [61]. In contrast, spatially-coupled LDPC codes derived from simple regular codes have been shown to universally achieve capacity for a variety of channels. However, there are also many open research problems concerning the practical implementation of spatially-coupled LDPC, see [62] for a recent overview. For example, the price to pay for the wave-like decoding behavior is a rate loss with respect to regular codes that are defined by the protograph  $\mathbf{P} = \sum_i \mathbf{P}_i$ .



---

## Generalized Product Codes

---

The practical implementation of BP decoding for LDPC codes at very high data rates poses a significant challenge. This motivates the use of coding schemes that are less complex (potentially sacrificing some performance). One particular example of such a coding scheme is discussed in this chapter, namely the use of GPCs in combination with iterative BDD.

We start in Sections 5.1 and 5.2 by reviewing GLDPC codes and PCs, respectively. GPCs can be regarded as a subclass of GLDPC codes and a formal definition is given in Section 5.3 together with several examples of GPCs. In Section 5.4, we discuss the assumed channel model and describe the decoding of GPCs via iterative BDD. Finally, in Section 5.5 we briefly outline and compare two approaches to perform an asymptotic DE analysis for GPCs.

## 5.1 Generalized Low-Density Parity-Check Codes

In the previous chapter, we have seen that the parity-check matrix  $\mathbf{H}$  of an LDPC code can be represented in terms of a bipartite Tanner graph where coded bits and parity-check equations are represented by VNs and CNs, respectively. An edge in the graph indicates if a certain bit participates in a certain parity-check equation (i.e., a row in  $\mathbf{H}$ ). The code can then be defined as the set of all VN bit assignments such that the parity-check equations corresponding to the CNs are satisfied.

This concept can be generalized by interpreting the CNs not just as simple parity-check equations but as component code constraints corresponding to smaller block codes (e.g., Hamming or BCH codes). In order to specify the code, one uses a binary matrix  $\mathbf{\Gamma} \in \{0, 1\}^{m \times n}$ . This matrix is interpreted as the adjacency matrix for an associated bipartite graph. The graph consists again of  $n$  VNs representing coded bits (one for each column in  $\mathbf{\Gamma}$ ) and  $m$  CNs representing component code constraints (one for each row in  $\mathbf{\Gamma}$ ). An edge between a VN and CN indicates if a certain bit participates in a certain constraint enforced by a component code. In addition to the matrix  $\mathbf{\Gamma}$  (or the corresponding graph), one also needs to specify  $m$  component codes  $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m$  that are associated with the  $m$  CNs in the graph. The overall code is defined as the set of all VN bit assignments that satisfy all component code constraints. The code thus defined is referred to as a GLDPC code.

Assuming that all component codes  $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m$  are linear codes, the resulting overall GLDPC code is also a linear code [12]. This implies that the code can alternatively be represented by using a parity-check matrix  $\mathbf{H}$  and a corresponding Tanner graph where CNs correspond again to simple parity-check equations. It should thus be stressed that the term “generalized” refers to the extended graphical representation by means of “generalized” CNs. The main reason for introducing these generalized CNs is that they add a layer of abstraction into the code representation. This may be helpful when constructing new codes or devising decoding algorithms. For example, assume that we have at our disposal an efficient decoding algorithm for some linear block code. The GLDPC code framework then allows us to build longer and potentially more powerful codes by using this block code as a building block. When decoding the overall code, we may take advantage of the available component code decoder, thereby allowing for an efficient overall decoding scheme.

## 5.2 Product Codes

PCs are one of the first examples that use the idea of building longer codes from shorter component codes [21]. In the following, we review the code construction and the representation as a GLDPC code.

Let  $\mathcal{B}$  be some binary linear block code of length  $n_{\mathcal{B}}$ . A PC is defined as the set of  $n_{\mathcal{B}} \times n_{\mathcal{B}}$  arrays such that each row and each column in the array is a codeword in  $\mathcal{B}$ .



As a side note, one subtlety that arises in the graphical representation of a GLDPC code is that, strictly speaking, the edges emanating from each CN should also be labeled with the corresponding component code bit positions [63]. If the component code has length  $n_B$  (i.e., the corresponding CN has degree  $n_B$ ), then in principle any of the  $n_B!$  possible permutations of the bit positions can be assigned to the edges. Choosing different assignments may result in an overall code with different properties, e.g., rate and minimum distance [63]. The reason this is not an issue for LDPC codes is that a parity-check equation is invariant under the order in which the bits appear in the equation. Moreover, for PCs and related code structures, the code is typically unambiguously defined by an accompanying array description.

## 5.3 Generalized Product Codes

With the background about GLDPC codes and PCs described in the previous two sections, we are now in the position to give a formal definition of a GPC. In particular, we adopt the convention in [64] and define a GPC as any GLDPC code whose Tanner graph representation consists exclusively of degree-2 VNs.<sup>1</sup> This implies that, similar to PCs, each coded bit is protected by exactly two component codes. However, the bits may not necessarily be arranged in the form of a rectangular array.

We remark that this terminology is nonstandard and GPCs are sometimes also referred to simply as “product-like” codes [9]. In the following, we review several examples of GPCs that are relevant for the appended papers.

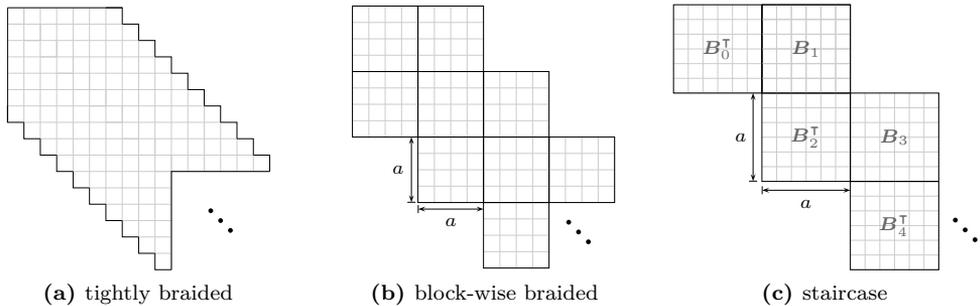
### 5.3.1 Braided Codes

Braided codes are proposed in [18] as “convolutional (or sliding) versions” of PCs. Similar to PCs, the code construction is based on a two-dimensional array where bits are placed in the array under the constraint that rows and columns form codewords in some component code. Depending on the type of component code, the resulting braided code is referred to either a braided block code or a braided convolutional code. In this thesis, we focus exclusively on the case where the component codes are block codes, and for simplicity we refer to the resulting codes simply as braided codes.

Braided codes come in different flavors, depending on the precise structure of the code array. Figs. 5.2(a) and (b) show two examples which are referred to as tightly braided codes and block-wise braided codes, respectively. In both cases, the array is conceptually infinite, i.e., one assumes the transmission of a continuous data stream. For tightly braided codes, the array structure consists of rows and columns that are shifted by one array element at each step. The simplified Tanner graph for a tightly braided code is shown for example in [18, Fig. 2(b)]. For the block-wise braided codes the array consists

---

<sup>1</sup>In [10], a slightly different definition of a GPC is given.



**Figure 5.2:** Differently shaped code arrays for the various GPCs discussed in Section 5.3.

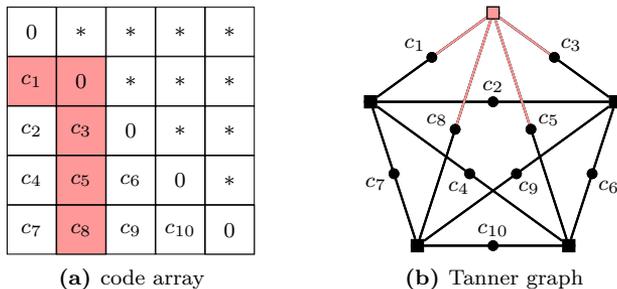
of three block ribbons with block size  $a$ . For example, the block size in Fig. 5.2(b) is given by  $a = 4$ .

Braided codes can be classified as spatially-coupled PCs or alternatively as convolutional PCs. (In [18], the term “GLDPC convolutional codes” is used instead.) Braided codes have been explicitly considered for the use in fiber-optical communication systems for example in [10]. The code construction we propose in Paper D includes block-wise braided codes as special cases, thereby enabling an asymptotic analysis for these codes.

### 5.3.2 Staircase Codes

Staircase codes are proposed in [9] as a new class of error-correcting codes for optical transport networks by “combining ideas from convolutional and block coding”. Given a component code  $\mathcal{B}$  of length  $n_{\mathcal{B}}$ , a staircase code is defined as the set of all matrix sequences  $\mathbf{B}_i \in \{0, 1\}^{a \times a}$ , where  $a = n_{\mathcal{B}}/2$  and  $i = 0, 1, 2, \dots$ , such that the rows in  $[\mathbf{B}_{i-1}^T \mathbf{B}_i]$  for all  $i \geq 1$  form valid codewords in  $\mathcal{B}$ . The matrix  $\mathbf{B}_0$  is assumed to be initialized to the all-zero matrix. The code array that corresponds to this definition has a characteristic staircase structure and is shown in Fig. 5.2(c), where  $n_{\mathcal{B}} = 12$  and  $a = 6$ . Similar to braided codes, staircase codes can also be classified as instances of spatially-coupled PCs.

When comparing staircase and braided codes, it should be mentioned that for braided codes, only soft-decision decoding of the component codes is studied in [18]. The authors in [9] consider this to be “unsuitable for high-speed fiber-optic communications”. Motivated by the excellent performance of staircase codes under iterative hard-decision decoding, the design of braided codes intended for fiber-optical communications is considered for example in [10]. The resulting braided code is found to be “competitive” to the staircase code designed in [65] suggesting that the performance of staircase and braided codes can be quite similar. This conclusion is also confirmed in Paper E, where we compare staircase codes and braided codes. For the considered parameters, both code classes perform almost identically in terms of waterfall performance and error floor.



**Figure 5.3:** Illustrations for an HPC with  $n_B = 5$ . In the array, “\*” means “equal to the transposed element”. The highlighted array elements illustrate one particular code constraint, which is also highlighted in the Tanner graph.

### 5.3.3 Half-Product Codes

Consider again the simplified Tanner graph representation of a PC shown in Fig. 5.1(b) which corresponds to a complete bipartite graph. The graph structure is a consequence of the array description of a PC which appears to be quite natural. On the other hand, in [66], Justesen points out that if one focuses instead on the graph, “it is not clear why a bipartite graph is preferable” and that in this case “the more natural concept [...] is a complete graph”. Such a complete Tanner graph indeed appears as one of the first examples in [63]. The resulting codes, however, have received very little attention in the literature thus far and, to the best of our knowledge, Justesen was the first to provide an interpretation of the graph structure in terms of a code array [66, Sec. III-B]. He also found a direct connection to conventional PCs which we briefly review in the following based on the descriptions in [67, Sec. IV-B].

Consider a conventional  $n_B \times n_B$  PC based on a component code  $\mathcal{B}$  of length  $n_B$ . Based on this PC, a new code is formed by keeping only codeword arrays that have zeros on the diagonal and are symmetric, in the sense that the array is equal to its transpose. Since the diagonal and upper (or lower) triangular part of the array do not contain “useful” bits, they can be ignored so that the effective length of the resulting code is given by  $m = \binom{n_B}{2}$ . Such codes are referred to as half-product codes (HPCs) in [66]. The Tanner graph of an HPC corresponds exactly to a “complete Tanner graph” with  $n_B$  CNs where each CN is connected to all other CNs through a VN. As an example, Figs. 5.3(a) and (b) show the code array and Tanner graph for an HPC assuming that  $n_B = 5$ , where one particular code constraint is highlighted in red.

HPCs and PCs are compared for example in [64], where it is shown that HPCs can have larger normalized minimum distance than PCs. It is also possible to extend the above definition to other classes of GPCs, i.e., other array shapes. For example, for the arrays shown in Figs. 5.2(a) and (b), it poses no conceptual problem to enforce the additional constraint (additional to the usual row and column component code constraints) that

the array should be symmetric with a zero diagonal. The resulting codes belong to the class of symmetric GPCs [64]. In general, symmetric GPCs use symmetry to reduce the block length of a GPC while employing the same component code [64].

## 5.4 Iterative Bounded-Distance Decoding

We assume that the intended “target” channel for GPCs is the binary symmetric channel (BSC) where each bit is flipped independently of all other bits with a certain crossover probability  $p$ . In the context of fiber-optical communications, this channel can be motivated by considering PM-QPSK transmission (i.e., independent binary modulation in both quadratures and polarizations) in combination with a minimum-distance detector that provides a hard decision about the transmitted signal. The BSC can be shown to be exact in the case where nonlinear transmission effects are ignored.

For the BSC, there exist very efficient algebraic BDD for several linear block codes, e.g., BCH codes. BDD corrects all error patterns with Hamming weight up to the error-correcting capability  $t$  of the code. The idea is then to use such codes as component codes for a GPC and decode the overall code by iteratively performing BDD of all component codes. While this decoding scheme is suboptimal, it has been shown to offer excellent performance provided that the code rate of the GPC is relatively high. For example, the staircase code designed in [9] has rate  $R = 239/255 \approx 0.937$  and performs only about 0.56 dB away from the channel capacity of the BSC under iterative BDD. Moreover, the decoder data flow requirements in this case are estimated to be more than two orders of magnitude smaller compared to the requirements for a comparable LDPC code with message-passing decoding [9].

The main conceptual problem that arises in the theoretical analysis of iterative BDD for GPCs over the BSC is that the component decoders may miscorrect in which case a successful (component) decoding is declared but the found codeword is not the correct one. Such miscorrections introduce additional bit errors into the iterative decoding process which makes a rigorous analysis challenging. One approach to avoid this issue is to ignore miscorrections entirely and assume the use of so-called idealized BDD. Such a decoder either outputs the correct codeword or declares a decoding failure. The assumption of idealized BDD over the BSC is conceptually equivalent to transmission over the BEC. For the BEC, each bit is erased independently of all other bits with a certain erasure probability  $p$ . In that case, the error-correcting capability  $t$  of the component code is interpreted as the erasure-correcting capability. The BEC is used in Papers D–F in order to allow for a rigorous theoretical analysis.

## 5.5 Density Evolution

The purpose of this section is to discuss two different approaches to perform an asymptotic performance analysis for GPCs under iterative BDD assuming transmission over

the BEC. The first approach is based on an ensemble argument and uses the ideas and techniques discussed in [24,68]. The second approach directly targets a sequence of deterministically constructed GPCs and is based on the work in [69,70]. Here, we only give a high-level overview of the basic approach idea in both cases. The main goal of this section is to contrast the two approaches and discuss potential advantages and disadvantages.

As a side note, the reader should be aware that the term “density evolution” for the asymptotic analysis may be somewhat misleading. This is because the parameter of interest in this case turns out to correspond to a simple probability and not a density.

### 5.5.1 Code Ensembles

The first and most widely used approach to perform an asymptotic analysis for GPCs is to define a “suitable” code ensemble. In the following, we review one such ensemble which is taken from [71] as an illustrative example. Let  $\mathcal{B}$  be a component code of length  $n_{\mathcal{B}}$ . Assume that there are  $m$  CNs of degree  $n_{\mathcal{B}}$  (each corresponding to the component code  $\mathcal{B}$ ) and  $mn_{\mathcal{B}}/2$  VNs of degree 2. In order to construct the Tanner graph, it is helpful to imagine that there are  $mn_{\mathcal{B}}$  half-edges emanating from all the CNs and VNs, respectively. One particular code in the ensemble is defined by the Tanner graph obtained by connecting these half-edges using a uniform random permutation. The ensemble is defined as the set of all codes defined by all possible choices of permutations.

The above ensemble definition is conceptually not much different from the definition of the regular LDPC code ensemble [24], except that the graph consists of generalized CNs. The asymptotic scenario considers the limit  $m \rightarrow \infty$ , i.e., one increases the number of VNs and CNs in the graph, while using a fixed component code. The principal conclusions from [24] (see also [68]) can be paraphrased as follows.

1. Asymptotically, the fixed-depth neighborhood of a randomly chosen VN or CN in the Tanner graph becomes a tree with high probability.
2. Assuming that the neighborhood is tree-like, the analysis of the expected iterative decoding performance is drastically simplified to the extent that it can usually be accomplished “in closed form” using a recursive expression.
3. There exists a concentration phenomenon that ensures that with high probability, a particular code taken uniformly at random from the ensemble will have actual performance close to the expected decoding performance computed in the previous step.

By combining these three conclusions, one can then study and analyze the asymptotic performance of the GPC ensemble defined above.

This ensemble approach appears to be appealing at first. For instance, one may be interested in using a fixed component code  $\mathcal{B}$  in practice (e.g., a triple-error correcting BCH code of length  $n_{\mathcal{B}} = 1023$ ) and the ensemble analysis enables an asymptotic analysis

for precisely this component code. Moreover, the approach is not limited to the analysis of iterative BDD but can be used to analyze the ensemble performance for a variety of different channels and iterative decoding algorithms [24].

On the other hand, for certain applications, the ensemble approach may not be appropriate. In particular, assume that we are interested in implementing a GPC with a fixed deterministic structure, e.g., a PC. In that case, it is not clear to what extent the ensemble approach is useful. In order to illustrate this, note that a PC is contained in the ensemble defined above for  $m = 2n_{\mathcal{B}}$  and a particular interleaver permutation. However, the ensemble approach only makes a statement about codes that are sampled uniformly at random from the ensemble and not particular ones. It is therefore not clear if the expected ensemble performance is somehow indicative for the performance of a PC.

### 5.5.2 Deterministic Codes

For sequences of *deterministic* GPCs, choosing a meaningful asymptotic scenario is not straightforward. For simplicity, let us restrict ourselves to “square” PCs, i.e., the case where both the row and column component codes have length  $n_{\mathcal{B}}$ . For the asymptotic scenario, we consider sequences of PCs with increasing array size. Increasing the array size, however, has *two* consequences. First, it leads to an increase in the number of component codes and thereby an increase in the number of CNs in the underlying Tanner graph. This is similar to the ensemble approach described in the previous subsection. However, increasing the array size also changes the component codes. In particular, the length  $n_{\mathcal{B}}$  of each component code does not remain fixed but it increases. This is different from the ensemble approach where the component code properties (including the length) are assumed to remain fixed.

When dealing with sequences of component codes with increasing lengths, one should also specify what happens to the other component code parameters, in particular the erasure-correcting capability  $t$ . There are essentially two possibilities that have been studied in the literature before. In the following, we briefly review both cases.

In the first case, one assumes that the erasure-correcting capability increases linearly with  $n_{\mathcal{B}}$ , i.e.,  $t$  is assumed to be a function of  $n_{\mathcal{B}}$ . In particular, one may assume that each component code can correct a fixed fraction  $\alpha \in (0, 1)$  of erasures in terms of its block length, i.e., we have  $t = \alpha n_{\mathcal{B}}$ . This case has been studied in [69]. The analysis is based on Chernoff bounds and the conclusion is quite simple: If one has access to component codes with  $t = \alpha n_{\mathcal{B}}$ , then, asymptotically, it is pointless to construct a PC out of these component codes since both the PC and each component code can operate reliably if (and only if) the erasure probability satisfies  $p < \alpha$ . In other words, the product construction is useless in this asymptotic scenario.

In the second case, one assumes that the erasure-correcting capability remains fixed. This is reasonable from a practical perspective because the complexity of algebraic BDD for BCH codes increases drastically with  $t$ . With this assumption, however, one finds that for any fixed erasure probability  $p$ , the decoding will fail with high probability in

the limit  $n_{\mathcal{B}} \rightarrow \infty$ . The reason for this is simple. Even if we choose  $p$  very small, the expected number of erasures per row and column in the PC array grows linearly with  $n_{\mathcal{B}}$  and will eventually far exceed the assumed finite erasure-correcting capability  $t$  of each component code.

In summary, both cases lead to somewhat unsatisfactory answers. The important thing to realize in the second case, however, is that we are essentially considering sequences of PCs with code rate approaching 1 as  $n_{\mathcal{B}} \rightarrow \infty$ . Due to this vanishing redundancy, it should not come as a surprise that operating at any fixed erasure probability is futile asymptotically. With this in mind, a meaningful asymptotic analysis can be performed by allowing the erasure probability to decay slowly as  $c/n_{\mathcal{B}}$  for some fixed constant  $c > 0$ . In other words, we are not considering a fixed channel anymore but the channel is changing in accordance with the strength of the PC. This approach is pioneered for PCs in [69, 70] and it heavily relies on properties from random graph theory. In Paper D, this approach is extended to a large class of deterministic GPCs based on properties of so-called inhomogeneous random graphs [26].

---

## Conclusions and Future Work

---

In this chapter, we summarize the main conclusions from the appended papers and outline some potentially interesting ideas for future work. The conclusions are structured in terms of the two topics that are investigated in this thesis.

### **Bit Mapper Optimization for Spectrally-Efficient Systems (Papers A and B)**

In Papers A and B, we study coded transmission systems that operate at high spectral efficiency over fiber-optical links without inline dispersion compensation. Assuming a linear coherent receiver, the classical AWGN channel with a modified SNR expression is used as a design channel.

In Paper A, we propose a method to optimize the bit mapper that determines the allocation of the coded bits from the channel encoder to the modulation or labeling bits of the signal constellation. The proposed method applies to an arbitrary protograph-based LDPC code. Compared to previous approaches for protograph-based codes, we use a fractional allocation between the modulation bits and the VN classes in the protograph. This allows for an unrestricted matching between different protographs and modulation formats. We also discuss the bit mapper optimization for spatially-coupled LDPC codes that are based on protographs and decoded using a windowed decoder. Our results show that by using an optimized bit mapper, the transmission reach can be extended by up to 8%, with almost no added system complexity. We also provide a simulative verification for a nonlinear transmission scenario based on the SSFM which confirms the accuracy of the assumed channel model.

In Paper B, we consider the bit mapper optimization for spatially-coupled codes in

more detail. In particular, we consider both spatially-coupled LDPC codes based on protographs and the spatially-coupled PC ensemble in [27]. In the first case, standard BP decoding is assumed, while in the second case we assume an iterative hard-decision decoding algorithm which is significantly less complex. For both cases, we consider terminated and tail-biting spatially-coupled codes. In general, for terminated spatially-coupled codes with long spatial length, the bit mapper optimization only results in marginal performance improvements, suggesting that a sequential or random allocation is close to optimal. On the other hand, an optimized allocation can significantly improve the performance of tail-biting spatially-coupled codes. Such codes do not possess an inherent termination boundary. In this case, the unequal error protection offered by the modulation bits can be used to create an artificial termination boundary that induces a wave-like decoding behavior. Unlike for terminated spatially-coupled codes, the wave effect in this case does not come at the price of a rate loss: tail-biting spatially-coupled codes have the same design rate as the underlying uncoupled codes.

An interesting direction for future work could be to study the bit mapper optimization for irregular spatially-coupled LDPC codes. In this thesis, we have assumed the use of spatially-coupled LDPC codes that are based on regular LDPC codes. While such codes are capacity-achieving, it has been shown in [72] that irregular spatially-coupled LDPC codes can offer some advantages when taking into consideration practical restrictions for parameters, e.g., a limited number of decoding iterations. For the irregular case, there is an opportunity to optimize the bit allocation by only considering the different VN degrees at one spatial position instead of considering the entire code chain. An optimized bit mapper allocation could for example lead to a decreased number of decoding iterations that are necessary to reach a certain target error rate.

### **Analysis and Design of Deterministic Generalized Product Codes (Papers C–F)**

In Paper C, we study parameter optimization for staircase codes assuming an extrinsic iterative hard-decision decoding algorithm. The optimization is based on a DE analysis for a spatially-coupled PC ensemble that shares structural similarities with staircase codes. Using this approach, staircase code parameters can be found at a significantly reduced optimization time compared to a simulation-based approach. We also propose an extension of staircase codes by allowing for multiple code constraints per row and column in the corresponding array description. This construction leads to larger staircase block sizes and steeper waterfall performance that better matches the predicted DE performance.

The optimization approach in Paper C is, however, only heuristically motivated. In particular, the DE analysis does not directly apply to staircase codes. This issue is addressed in Paper D, where we consider an asymptotic DE analysis for deterministic GPCs with a fixed Tanner graph structure. The main conclusion from Paper D can be summarized as follows. There exists a large class of deterministic GPCs for which a rigorous asymptotic DE analysis assuming iterative BDD over the BEC is possible.

---

For example, the proposed code construction and analysis can be used to study the asymptotic performance of

- conventional PCs [21], staircase codes [65], and block-wise braided codes [18],
- GPCs employing a mixture of different component codes with varying erasure-correcting capabilities such as irregular PCs [73, 74],
- symmetric GPCs [64] such as HPCs [66] which can be seen as symmetric subcodes of certain GPCs.

The DE analysis in Paper D can also be used to study different decoding schedules that are practically relevant such as row/column decoding for PCs or windowed decoding of staircase and braided codes.

In Papers E and F, we use the proposed code construction and DE analysis to study some relevant classes of GPCs in more detail. In Paper E, we provide a comparison between staircase codes, block-wise braided codes, and the symmetric subcode of a block-wise braided code which is referred to as a half-braided code. Our results indicate that half-braided codes can outperform both staircase codes and braided codes in the waterfall performance, at a lower error floor and decoding delay.

In Paper F, we consider spatially-coupled PCs in more detail. In particular, we revisit the spatially-coupled PC ensemble that is used in Paper C for the parameter optimization of staircase codes. It is shown in [27] that this ensemble can approach the capacity of the BSC at high rates. Motivated by this result, our main goal is to compare the ensemble performance to the performance of deterministic GPCs with a spatially-coupled structure via their respective DE equations. For the BEC, it is shown that there exists a family of deterministic braided codes that performs asymptotically equivalent to the ensemble. It is also shown that there exists a related but structurally simpler family of braided codes with essentially the same performance, even though the DE equations are not exactly equivalent. Lastly, we consider spatially-coupled PCs with a mixture of different component codes. In that case, the conclusion is that employing such component code mixtures for spatially-coupled PCs is not beneficial from an asymptotic point of view.

In the following, we suggest two potentially interesting topics for future work. The first topic concerns the asymptotic performance of deterministic GPCs over the BSC. While the obtained results for the BEC can be used to approximate the code performance over the BSC, it would be desirable to rigorously characterize the BSC performance including the effect of decoder miscorrections. For example, the equivalence between the spatially-coupled PC ensemble in [27] and the deterministic spatially-coupled PCs in Paper F only holds for the BEC. This is not sufficient to show that the same deterministic codes are also capacity-achieving over the BSC. On the other hand, the proof in [27] relies partially on the fact that the impact of miscorrections becomes small if the error-correction capability increases. Similar conclusions should also hold for deterministic GPCs, at least qualitatively.

Another potentially interesting topic is the investigation of the finite-length scaling behavior of deterministic GPCs, similar to the scaling analysis for certain LDPC code ensembles presented in [75]. A generalization to deterministic GPCs may give a relatively complete picture of the performance of these codes under iterative decoding.

---

## Bibliography

---

- [1] K. F. Rauscher, “ROGUCCI study final report,” IEEE Communications Society, Tech. Rep., 2010.
- [2] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, “Capacity limits of optical fiber networks,” *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, Feb. 2010.
- [3] E. Agrell and M. Karlsson, “Satellite constellations: Towards the nonlinear channel capacity,” in *Proc. IEEE Photon. Conf.*, Burlingame, CA, 2012.
- [4] A. D. Ellis, J. Zhao, and D. Cotter, “Approaching the non-linear Shannon limit,” *J. Lightw. Technol.*, vol. 28, no. 4, pp. 423–433, Feb. 2010.
- [5] B. P. Smith and F. R. Kschischang, “Future prospects for FEC in fiber-optic communications,” *IEEE J. Sel. Topics. Quantum Electron.*, vol. 16, no. 5, pp. 1245–1257, Oct. 2010.
- [6] L. Schmalen, A. J. de Lind van Wijngaarden, and S. ten Brink, “Forward error correction in optical core and optical access networks,” *Bell Labs Tech. J.*, vol. 18, no. 3, pp. 39–66, Mar. 2013.
- [7] L. Beygi, E. Agrell, J. M. Kahn, and M. Karlsson, “Coded modulation for fiber-optic networks,” *IEEE Signal Processing Mag.*, vol. 31, no. 2, pp. 93–103, Mar. 2014.
- [8] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, Jul. 1948.
- [9] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase codes: FEC for 100 Gb/s OTN,” *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

- [10] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Mazahreh, "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication," in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.
- [11] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.
- [12] W. Ryan and S. Lin, *Channel Codes Classical and Modern*. Cambridge University Press, 2009.
- [13] A. J. Felström and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
- [14] M. Lentmaier, A. Sridharan, K. S. Zigangirov, and D. J. Costello, Jr., "Terminated LDPC convolutional codes with thresholds close to capacity," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Adelaide, Australia, 2005.
- [15] R. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1963.
- [16] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [17] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions," in *Proc. 7th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Gothenburg, Sweden, 2012.
- [18] A. J. Felström, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, "Braided block codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jul. 2009.
- [19] E. Agrell and M. Karlsson, "Power-efficient modulation formats in coherent transmission systems," *J. Lightw. Technol.*, vol. 27, no. 22, pp. 5115–5126, Nov. 2009.
- [20] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progress Report 42-154, JPL*, 2005.
- [21] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Apr. 1954.
- [22] "Forward error correction for high bit-rate DWDM submarine systems," ITU-T Recommendation G.975.1, 2004.
- [23] L. M. Zhang and F. R. Kschischang, "Staircase codes with 6% to 33% overhead," *J. Lightw. Technol.*, vol. 32, no. 10, pp. 1999–2002, May 2014.

- 
- [24] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [25] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high-rates with iterative hard-decision decoding,” *arxiv:1202.6095v3 [cs.IT]*, May 2015. [Online]. Available: <http://arxiv.org/abs/1202.6095.pdf>
- [26] B. Bollobás, S. Janson, and O. Riordan, “The phase transition in inhomogeneous random graphs,” *Random Structures and Algorithms*, vol. 31, no. 1, pp. 3–122, Aug. 2007.
- [27] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high rates with iterative hard-decision decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [28] G. P. Agrawal, *Nonlinear Fiber Optics*, 4th ed. Academic Press, 2006.
- [29] M. Secondini and E. Forestieri, “The nonlinear Schrödinger equation in fiber-optic systems,” *Riv. Mat. Univ. Parma*, vol. 8, pp. 69–98, Sep. 2008.
- [30] O. V. Sinkin, R. Holzlohner, J. Zweck, and C. R. Menyuk, “Optimization of the split-step Fourier method in modeling optical-fiber communications systems,” *J. Lightw. Technol.*, vol. 21, no. 1, pp. 61–68, Jan. 2003.
- [31] G. P. Agrawal, *Fiber-optic Communication Systems*, 4th ed. Wiley-Interscience, 2010.
- [32] A. Mecozzi, “Limits to long-haul coherent transmission set by the Kerr nonlinearity and noise of the in-line amplifiers,” *J. Lightw. Technol.*, vol. 12, no. 11, pp. 1993–2000, Nov. 1994.
- [33] M. I. Yousefi, “Information transmission using the nonlinear Fourier transform,” Ph.D. dissertation, University of Toronto, 2013.
- [34] L. Beygi, “Channel-aware multilevel coded modulation for coherent fiber-optic communications,” Ph.D. dissertation, Chalmers University of Technology, 2013.
- [35] L. Beygi, E. Agrell, P. Johannisson, M. Karlsson, and H. Wymeersch, “A discrete-time model for uncompensated single-channel fiber-optical links,” *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3440–3450, Nov. 2012.
- [36] A. Carena, V. Curri, G. Bosco, P. Poggiolini, and F. Forghieri, “Modeling of the impact of nonlinear propagation effects in uncompensated optical coherent transmission links,” *J. Lightw. Technol.*, vol. 30, no. 10, pp. 1524–1539, May 2012.

- [37] P. Poggiolini, A. Carena, V. Curri, G. Bosco, and F. Forghieri, “Analytical modeling of nonlinear propagation in uncompensated optical transmission links,” *IEEE Photon. Technol. Lett.*, vol. 23, no. 11, pp. 742–744, Jun. 2011.
- [38] P. Johannisson, “Analytical modeling of nonlinear propagation in a strongly dispersive optical communication system,” *arXiv:1205.2193v2 [physics.optics]*, May 2012. [Online]. Available: <http://arxiv.org/abs/1205.2193>
- [39] C. E. Shannon, “Communication in the presence of noise,” *Proc. IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- [40] G. D. Forney, Jr. and G. Ungerboeck, “Modulation and coding for linear Gaussian channels,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.
- [41] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.
- [42] A. Bennatan and D. Burshtein, “Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.
- [43] U. Wachsmann, R. Fischer, and J. Huber, “Multilevel codes: theoretical concepts and practical design rules,” *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, Jul. 1999.
- [44] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [45] D. J. Costello and G. D. Forney, Jr., “Channel coding: The road to channel capacity,” *Proc. IEEE*, vol. 95, no. 6, pp. 1150–1177, Jun. 2007.
- [46] B. Smith and F. R. Kschischang, “A pragmatic coded modulation scheme for high-spectral-efficiency fiber-optic communications,” *J. Lightw. Technol.*, vol. 30, no. 13, pp. 2047–2053, Jul. 2012.
- [47] I. B. Djordjevic, M. Arabaci, and L. L. Minkov, “Next generation FEC for high-capacity communication in optical transport networks,” *J. Lightw. Technol.*, vol. 27, no. 16, pp. 3518–3530, Aug. 2009.
- [48] L. Szczecinski and A. Alvarado, *Bit-Interleaved Coded Modulation: Fundamentals, Analysis, and Design*, 1st ed. Wiley-Interscience, 2014.
- [49] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, “Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003.

- 
- [50] G. Richter, A. Hof, and M. Bossert, "On the mapping of low-density parity-check codes for bit-interleaved coded modulation," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Nice, Italy, 2007.
- [51] T. Cheng, K. Peng, J. Song, and K. Yan, "EXIT-aided bit mapping design for LDPC coded modulation with APSK constellations," *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 777–780, Jun. 2012.
- [52] R. Liu, P. Spasojevic, and E. Soljanin, "Reliable channel regions for good binary codes transmitted over parallel channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1405–1424, Apr. 2006.
- [53] I. Sason and I. Goldenberg, "Coding for parallel channels: Gallager bounds and applications to turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 7, pp. 2394–2428, Jul. 2007.
- [54] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [55] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [56] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.
- [57] S. Kudekar, T. Richardson, and R. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [58] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "AWGN channel analysis of terminated LDPC convolutional codes," in *Proc. Information Theory and Applications Workshop (ITA)*, La Jolla, CA, 2011.
- [59] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [60] M. G. Luby, M. Mitzenmacher, M. A. Shokrollah, and D. A. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proc. AMC Symp. on Theory of Computing (STOC)*, New York, USA, 1998.
- [61] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

- [62] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliever, D. G. M. Mitchell, and R. Smarandache, “Spatially coupled sparse codes on graphs – theory and practice,” *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.
- [63] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
- [64] H. D. Pfister, S. K. Emmadi, and K. Narayanan, “Symmetric product codes,” in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, 2015.
- [65] B. P. Smith, “Error-correcting codes for fibre-optic communication systems,” Ph.D. dissertation, University of Toronto, 2011.
- [66] J. Justesen, “Performance of product codes and related structures with iterated decoding,” *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.
- [67] C. Häger, H. D. Pfister, A. Graell i Amat, F. Brännström, and E. Agrell, “A deterministic construction and density evolution analysis for generalized product codes,” in *Proc. Int. Zurich Seminar on Communications (IZS)*, Zürich, Switzerland, 2016.
- [68] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, “Analysis of random processes via and-or tree evaluation,” in *Proc. 9th Annual ACM-SIAM Symp. Discrete Algorithms*, San Francisco, CA, 1998.
- [69] M. Schwartz, P. Siegel, and A. Vardy, “On the asymptotic performance of iterative decoders for product codes,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Adelaide, SA, 2005.
- [70] J. Justesen and T. Høholdt, “Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes,” in *Proc. IEEE Information Theory Workshop (ITW)*, Tahoe City, CA, 2007.
- [71] Y.-Y. Jian, “On the analysis of spatially-coupled GLDPC codes and the weighted min-sum algorithm,” Ph.D. dissertation, Texas A&M University, 2013.
- [72] A. Leven and L. Schmalen, “Status and recent advances on forward error correction technologies for lightwave systems,” *J. Lightw. Technol.*, vol. 32, no. 16, pp. 2735–2750, Aug. 2014.
- [73] S. Hirasawa, M. Kasahara, Y. Sugiyama, and T. Namekawa, “Modified product codes,” *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 299–306, Mar. 1984.
- [74] M. Alipour, O. Etesami, G. Maatouk, and A. Shokrollahi, “Irregular product codes,” in *Proc. IEEE Information Theory Workshop (ITW)*, Lausanne, Switzerland, 2012.
- [75] A. Amraoui, A. Montanari, T. J. Richardson, and R. L. Urbanke, “Finite-length scaling for iteratively decoded LDPC ensembles,” *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.

**Part II**

**Papers**



PAPER **A**

**Improving Soft FEC Performance for Higher-Order Modulations via  
Optimized Bit Channel Mappings**

Christian Häger, Alexandre Graell i Amat, Fredrik Brännström,  
Alex Alvarado, and Erik Agrell

*Optics Express*, vol. 22, no. 12, pp. 14544–14558,  
June 2014

*The layout has been revised.*

## Abstract

Soft forward error correction with higher-order modulations is often implemented in practice via the pragmatic bit-interleaved coded modulation paradigm, where a single binary code is mapped to a nonbinary modulation. In this paper, we study the optimization of the mapping of the coded bits to the modulation bits for a polarization-multiplexed fiber-optical system without optical inline dispersion compensation. Our focus is on protograph-based low-density parity-check (LDPC) codes which allow for an efficient hardware implementation, suitable for high-speed optical communications. The optimization is applied to the AR4JA protograph family, and further extended to protograph-based spatially coupled LDPC codes assuming a windowed decoder. Full field simulations via the split-step Fourier method are used to verify the analysis. The results show performance gains of up to 0.25 dB, which translate into a possible extension of the transmission reach by roughly up to 8%, without significantly increasing the system complexity.

## 1 Introduction

There is currently a large interest in developing practical coded modulation (CM) schemes that can achieve high spectral efficiency close to the ultimate capacity limits of optical fibers [1]. Pragmatic bit-interleaved coded modulation (BICM) in combination with low-density parity-check (LDPC) codes is one of the most popular capacity-approaching CM techniques for achieving high spectral efficiency, due to its simplicity and flexibility [2]. For a BICM system, a helpful abstraction is to think about transmitting data using a single forward error correction (FEC) encoder over a set of parallel binary-input channels, or simply bit channels, with different qualities. This is due to the fact that bits are not protected equally throughout the signal constellation. With this useful picture, an immediate problem is how to best allocate the coded bits from the encoder to these channels. As a baseline, a random or consecutive/sequential mapping is commonly used in practice. However, by optimizing the mapping strategy, one can improve the system performance, at almost no increased complexity cost. While BICM has been studied for fiber-optical communications by many authors, see e.g., [3] or [4] and references therein, to the best of our knowledge, optimized bit channel mappings have not yet been studied for such systems. In the following, we use the term “bit mapper” to denote the device that performs the bit channel mapping. We remark that other terms, e.g., “bit interleaver” or “mapping device”, are also frequently used in the literature.

In this paper, we address the bit mapper optimization for a BICM system based on LDPC codes in the context of long-haul fiber-optical communications. Our target

system operates over a communication link with a lumped amplification scheme and without optical inline dispersion compensation. In general, the signal undergoes a complicated evolution and interacts with amplified spontaneous emission (ASE) noise and co-propagating signals through dispersive and nonlinear effects. For dispersion uncompensated transmission, it has been shown that an additive Gaussian noise (GN) model can be assumed, provided that dispersive effects are dominant and nonlinear effects are weak [5, 6]. We use the GN model for our analysis, which accounts for both the ASE noise from inline erbium-doped fiber amplifiers (EDFAs) and nonlinear noise due to the optical Kerr effect.

The starting point for the optimization problem is a fixed modulation format and a given error correction code, i.e., we do not consider the *joint* design of the modulation, bit mapper, and code. This scenario is often encountered in practice when the modulation and code have been designed separately and/or are predetermined according to some communication standard. Our focus is on protograph-based LDPC codes [7], which are very attractive from a design perspective and allow for a high-speed hardware implementation, suitable for fiber-optical communications [8]. A protograph is a (small) bipartite graph, from which the Tanner graph defining the code is obtained by a copy-and-permute procedure. As one illustrative example for protograph-based codes, we consider the AR4JA protographs developed by researchers from JPL/NASA in [9]. We also consider bit mapper optimization for protograph-based spatially-coupled low-density parity-check (SC-LDPC) codes using the windowed decoder (WD) proposed in [10]. SC-LDPC codes, originally introduced as LDPC convolutional codes in [11], have attracted a lot of attention due to their capacity-achieving performance under belief propagation (BP) decoding for a variety of communication channels [12]. SC-LDPC codes can be constructed using protographs and they are considered as viable candidates for future spectrally efficient fiber-optical systems [8].

Most of the literature about bit mapper optimization deals with irregular LDPC codes that are not based on protographs, see e.g., [13, 14]. Attempts to improve the performance of BICM systems with protograph-based codes through bit mapper optimization have been previously made in [15–17]. In [15], a mapping strategy inspired by the water-filling algorithm for parallel channels called variable degree matched mapping (VDMM) is presented. This idea is extended in [16], where the authors exhaustively search over all possible nonequivalent connections between protograph nodes and modulation bits showing performance improvements over VDMM. As pointed out in [17], the above approaches are somewhat restrictive in the sense that only certain protographs can be used with certain modulation formats. A more flexible approach is proposed in [17], which is in principle suitable for any protograph structure and modulation but relies on a larger intermediate protograph.

Our optimization of the bit mapper is based on the decoding threshold over the additive white Gaussian noise (AWGN) channel similar to, e.g., [13, 14, 16], albeit assuming a fixed number of decoding iterations. The decoding threshold divides the channel qual-

ity parameter range (in our case the equivalent signal-to-noise ratio (SNR) of the GN model) into a region where reliable decoding is possible and where it is not. In the asymptotic case, i.e., assuming infinite codeword length, density evolution (DE) or one-dimensional simplifications via extrinsic information transfer (EXIT) functions can be used to find the decoding threshold for LDPC codes under BP decoding [18]. Approximate decoding thresholds of protograph-based codes assuming binary modulation can be obtained by using the protograph extrinsic information transfer (P-EXIT) analysis [19]. The approach proposed here relies on a modified P-EXIT analysis which allows for a fractional allocation between protograph nodes and modulation bits. This approach is, to the best of our knowledge, novel in the context of protograph-based codes and different from the approaches described in [15–17]. In particular, a fractional allocation allows for an unrestricted matching of protographs and modulation formats and additionally does not suffer from an increased design complexity due to a larger intermediate protograph. We also discuss several ways to reduce the optimization complexity. In particular, we introduce periodic bit mappers for SC-LDPC codes with a WD, which is based on the results we previously presented in [20], where optimized bit mappers are found for (nonprotograph-based) SC-LDPC codes assuming parallel binary erasure channels (BECs) without considering the WD. The use of a WD in this paper is motivated by the reduced complexity and decoding delay with respect to full decoding. Finally, we provide a simulative verification assuming both linear and nonlinear transmission scenarios. For the latter case, we use the split-step Fourier method (SSFM) to show that the performance improvements predicted from the AWGN analysis can be achieved for a realistic transmission scenario including nonlinear effects.

## 1.1 Notation

Vectors and matrices are typeset in bold font by lowercase letters  $\mathbf{a}$  and capital letters  $\mathbf{A}$ , respectively. Matrix transpose is denoted by  $(\cdot)^\top$ , Hermitian transpose by  $(\cdot)^\dagger$ , and the squared norm of a complex vector by  $\|\mathbf{a}\|^2$ .  $\mathbf{I}_n$  denotes the identity matrix of size  $n$ . Complex conjugation is denoted by  $(\cdot)^*$ .  $\delta(t)$  is Dirac's delta function, whereas  $\delta[k]$  is the Kronecker delta. Convolution is denoted by  $\otimes$ .  $\mathbb{N}_0$ ,  $\mathbb{R}$ , and  $\mathbb{C}$  denote the set of nonnegative integers, real numbers, and complex numbers, respectively. Random variables and vectors are denoted by capital letters and their realizations by lowercase letters. The probability density function (PDF) of a random variable  $Y$  conditioned on the realization of another random variable  $X$  is denoted by  $f_{Y|X}(y|x)$  and the expected value by  $\mathbb{E}[\cdot]$ .

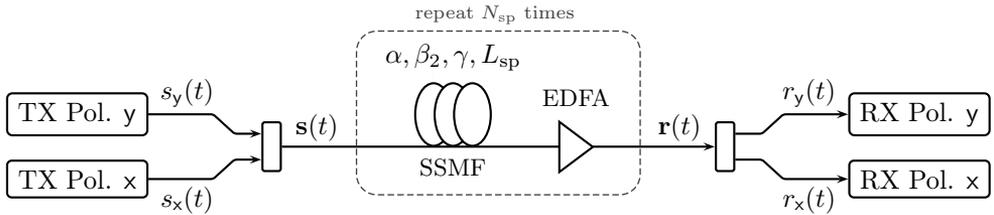


Figure 1: Block diagram of the consider fiber-optical transmission system.

## 2 System Model

### 2.1 Continuous-Time Channel

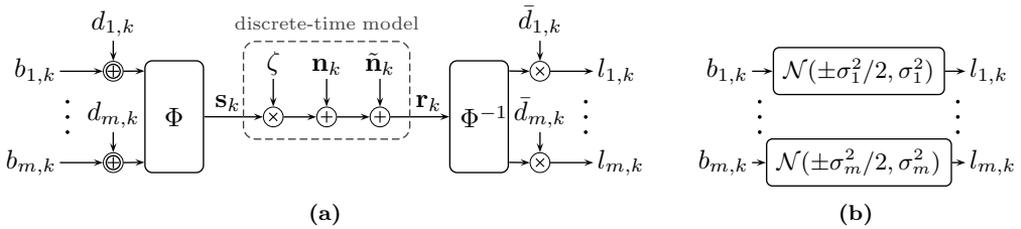
We consider transmission of a polarization-multiplexed (PM) signal over a standard single-mode fiber (SSMF) with a lumped amplification scheme as shown in Fig. 1. The optical link consists of  $N_{\text{sp}}$  spans of SSMF with length  $L_{\text{sp}}$ . The baseband signal in each polarization is generated via a linear pulse modulation according to  $s_{\text{x}}(t) = \sum_k s_{\text{x},k} p(t - k/R_s)$ , where  $s_{\text{x},k} \in \mathbb{C}$  are the information symbols,  $p(t)$  the real-valued pulse shape, and  $R_s$  the symbol rate. (We give expressions for polarization x only, if polarization y has an equivalent expression.) The PM signal  $\mathbf{s}(t) = (s_{\text{x}}(t), s_{\text{y}}(t))^{\text{T}}$  is launched into the fiber and propagates according to [21, Ch. 3]

$$\frac{\partial \mathbf{v}(t, z)}{\partial z} = -\frac{\alpha - g(z)}{2} \mathbf{v}(t, z) - j \frac{\beta_2}{2} \frac{\partial^2 \mathbf{v}(t, z)}{\partial t^2} + j \gamma \mathbf{v}(t, z) \|\mathbf{v}(t, z)\|^2 + \mathbf{w}(t, z), \quad (\text{A.1})$$

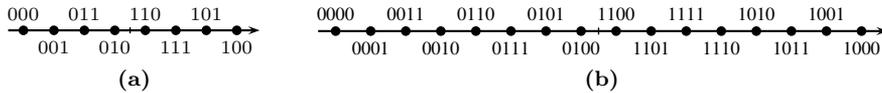
where  $\mathbf{v}(t, z)$  is the complex baseband representation of the electric field and the input to the first fiber span and the output signal are  $\mathbf{s}(t) = \mathbf{v}(t, 0)$  and  $\mathbf{r}(t) = \mathbf{v}(t, N_{\text{sp}} L_{\text{sp}})$ , respectively. In (A.1),  $\alpha$  is the attenuation coefficient,  $\beta_2$  the chromatic dispersion coefficient, and  $\gamma$  the nonlinear Kerr parameter. The terms  $g(z)$  and  $\mathbf{w}(t, z) = (w_{\text{x}}(t, z), w_{\text{y}}(t, z))^{\text{T}}$  model the amplifier gain and the generated ASE noise [22, p. 84]. Each EDFA introduces circularly symmetric complex Gaussian noise with two-sided power spectral density (PSD)  $N_{\ell} = (G - 1) h \nu_s n_{\text{sp}}$  [1, eq. (54)] per polarization, where  $G = e^{\alpha L_{\text{sp}}}$  is the amplifier gain,  $h$  is Planck's constant,  $\nu_s$  the carrier frequency, and  $n_{\text{sp}}$  the spontaneous emission factor. A standard coherent linear receiver is used, consisting of an equalizer, a pulse-matched filter and a symbol-time sampler. This amounts to  $r_{\text{x},k} = r_{\text{x}}(t) \otimes h(t) \otimes p(-t)|_{t=k/R_s}$ , where the frequency response of the equalizer  $h(t)$  is  $H(f) = \exp(j2\beta_2\pi^2 f^2 N_{\text{sp}} L_{\text{sp}})$ .

### 2.2 Discrete-Time Channel

An approximate discrete-time model for the received samples  $\mathbf{r}_k = (r_{\text{x},k}, r_{\text{y},k})^{\text{T}}$  based on the transmitted symbols  $\mathbf{s}_k = (s_{\text{x},k}, s_{\text{y},k})^{\text{T}}$  is given by  $\mathbf{r}_k \approx \zeta \mathbf{s}_k + \mathbf{n}_k + \tilde{\mathbf{n}}_k$ , where  $\zeta \in \mathbb{C}$  [5]. The term  $\mathbf{n}_k = (n_{\text{x},k}, n_{\text{y},k})^{\text{T}}$  accounts for the linear ASE noise with  $\mathbb{E}[\mathbf{N}_k \mathbf{N}_k^{\dagger}] =$



**Figure 2:** (a) BICM block diagram including the channel symmetrization technique. (b) Approximate model with parallel Gaussian LLR channels.



**Figure 3:** The considered signal constellations in each dimension.

$P_{\text{ASE}} \mathbf{I}_2 \delta[k - k']$ , where  $P_{\text{ASE}} = N_{\text{sp}} N_\ell R_s$ . The term  $\tilde{\mathbf{n}}_k = (\tilde{n}_{x,k}, \tilde{n}_{y,k})^\top$  accounts for nonlinear noise with  $\mathbb{E}[\tilde{\mathbf{N}}_k \tilde{\mathbf{N}}_{k'}^\dagger] = \eta P^3 \mathbf{I}_2 \delta[k - k']$ , where  $P = \lim_{T \rightarrow \infty} (\int_{-T}^T s_x(t)^2 dt) / (2T)$  is the transmit power per polarization (assumed to be equal for both polarizations).  $\eta$  is a function of the link parameters  $\alpha, \beta_2, \gamma, L_{\text{sp}}, N_{\text{sp}}$  and the symbol rate  $R_s$  [5, eq. (15)], and  $|\zeta|^2 = 1 - |\eta|P^2$ . The conditional PDF in this model is assumed to be Gaussian according to

$$f_{\mathbf{R}_k | \mathbf{S}_k}(\mathbf{r}_k | \mathbf{s}_k) = \frac{1}{(\pi P_N)^2} \exp\left(-\frac{\|\mathbf{r}_k - \zeta \mathbf{s}_k\|^2}{P_N}\right), \quad (\text{A.2})$$

where  $P_N = P_{\text{ASE}} + \eta P^3$ . The equivalent SNR is defined as  $\rho \triangleq |\zeta|^2 P / (P_{\text{ASE}} + \eta P^3)$ .

## 2.3 Bit-Interleaved Coded Modulation

The transmitted symbols  $\mathbf{s}_k$  in each time instant  $k$  take on values from a discrete signal constellation  $\mathcal{X} \subset \mathbb{C}^2$ . Each point in the constellation is labeled with a unique binary string of length  $m = \log_2 |\mathcal{X}|$ , where  $b_i(\mathbf{a})$ ,  $1 \leq i \leq m$ , denotes the  $i$ th bit in the binary string assigned to  $\mathbf{a} \in \mathcal{X}$  (counting from left to right). Consider now the block diagram shown in Fig. 2(a), where the modulo 2 addition of  $d_{i,k}$  and multiplication by  $\bar{d}_{i,k} = (-1)^{d_{i,k}}$  is explained further below and can be ignored for now. At each time instant  $k$ , the modulator  $\Phi$  takes  $m$  bits  $b_{i,k}$ ,  $1 \leq i \leq m$ , and maps them to one of the constellation points according to the binary labeling. We consider two product constellations of one-dimensional constellations labeled with the binary reflected Gray code (BRGC) as shown in Fig. 3, which we refer to as PM-64-QAM and PM-256-QAM. At the receiver, the demodulator  $\Phi^{-1}$  computes soft reliability information about the

transmitted bits in the form of the log-likelihood ratios (LLRs)

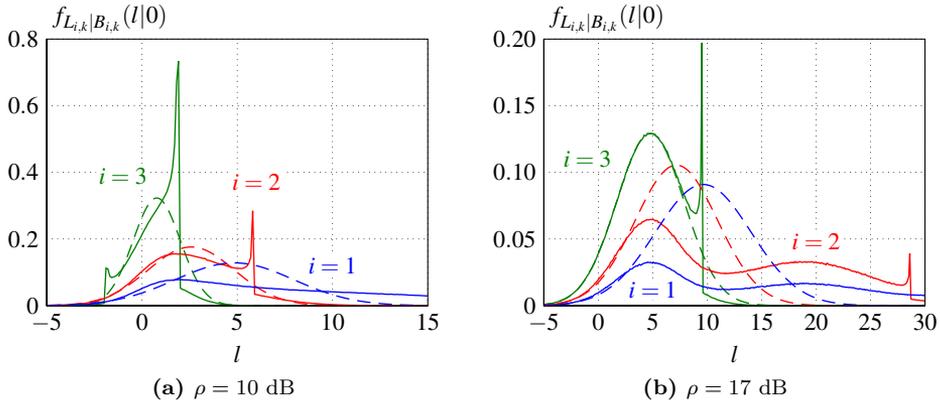
$$l_{i,k} \triangleq \log \frac{f_{\mathbf{R}_k|B_{i,k}}(\mathbf{r}_k|0)}{f_{\mathbf{R}_k|B_{i,k}}(\mathbf{r}_k|1)} = \log \frac{\sum_{\mathbf{s} \in \mathcal{X}_{i,0}} f_{\mathbf{R}_k|\mathbf{S}_k}(\mathbf{r}_k|\mathbf{s})}{\sum_{\mathbf{s} \in \mathcal{X}_{i,1}} f_{\mathbf{R}_k|\mathbf{S}_k}(\mathbf{r}_k|\mathbf{s})}, \quad (\text{A.3})$$

where  $\mathcal{X}_{i,u} \triangleq \{\mathbf{a} \in \mathcal{X} : b_i(\mathbf{a}) = u\}$  is the subconstellation where all points have the bit  $u$  at the  $i$ th position of their binary label.

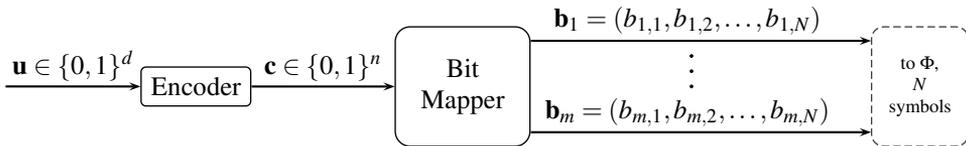
A useful way to think about the setup depicted in Fig. 2(a) is to imagine transmitting over a set of parallel bit channels, where one may interpret the conditional distribution of the LLR  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  as a bit channel. In the following, we say that a bit channel  $f_{L|B}(l|b)$  is symmetric if  $f_{L|B}(l|0) = f_{L|B}(-l|1)$  and the channel is referred to as an LLR channel if  $f_{L|B}(l|0)e^l = f_{L|B}(l|1)$ . One can show that  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  is an LLR channel, but not necessarily symmetric in general. Symmetry can be enforced by adding modulo 2 independent and identically distributed bits  $d_{i,k}$  to the bits  $b_{i,k}$  and multiplying the corresponding LLR by  $\bar{d}_{i,k}$  (see Fig. 2(a)) [23]. The symmetry condition is an important requirement for the analysis in Section 3.3, where one implicitly relies on the assumption that the all-zero codeword has been transmitted [24, p. 389].

To simplify the analysis, the original bit channels are replaced with parallel symmetric Gaussian LLR channels, as shown in Fig. 2(b), where an LLR channel  $f_{L|B}(l|b)$  is called a symmetric Gaussian LLR channel with parameter  $\sigma^2$  if  $L \sim \mathcal{N}(\sigma^2/2, \sigma^2)$  conditioned on  $B = 0$  and  $L \sim \mathcal{N}(-\sigma^2/2, \sigma^2)$  conditioned on  $B = 1$ . In order to find a correspondence between the LLR channels  $f_{L_{i,k}|B_{i,k}}(\cdot|\cdot)$  and the parameters  $\sigma_i^2$ , one may match the mutual information (MI) according to  $\sigma_i^2 = J^{-1}(I_i(\rho))^2$ , where  $I_i(\rho) = I(B_{i,k}; L_{i,k})$  is independent of  $k$  and  $J(\sigma)$  denotes the MI between the output of a symmetric Gaussian LLR channel and uniform input bits. As an example and to visualize the different bit channel qualities, in Fig. 4 we compare the LLR channels (solid lines, estimated via histograms) with the approximated Gaussian LLR channels (dashed lines) assuming an AWGN channel and two different values of  $\rho$  for the three distinct bit channels of PM-64-QAM (see Fig. 3(a)). It can be seen that the actual densities are clearly non-Gaussian. However, the Gaussian approximation is quite accurate for the bit mapper optimization as shown later and allows for a major simplification of the analysis, thereby justifying its use.

Consider now the case where a binary code  $\mathcal{C} \subset \{0, 1\}^n$  of length  $n$  and dimension  $d$  is employed and each codeword  $\mathbf{c} = (c_1, \dots, c_n)$  is transmitted using  $N = n/m$  symbols  $\mathbf{s}_k$ . The allocation of the coded bits to the modulation bits (i.e., the different bit channels in Fig. 2(b)) is determined by a bit mapper as shown in Fig. 5, where the vectors  $\mathbf{b}_1, \dots, \mathbf{b}_m$  are of length  $N$ . Our goal is to find good bit mappers for a fixed code and modulation. As a baseline, we consider a consecutive mapper according to  $b_{i,k} = c_{(k-1)m+i}$  for  $1 \leq i \leq m$ ,  $1 \leq k \leq N$ .



**Figure 4:** Comparison of the LLR channels for PM-64-QAM including channel symmetrization (solid lines) with the Gaussian LLR channels that have the same MI (dashed lines).



**Figure 5:** Block diagram illustrating the purpose of the bit mapper.

### 3 Protograph-Based LDPC Codes

An LDPC code of length  $n$  and dimension  $d$  is defined via a sparse parity-check matrix  $\mathbf{H} = [h_{i,j}] \in \{0, 1\}^{c \times n}$ , where  $c = n - d$ . There exist different methods to construct “good” LDPC codes, i.e., good matrices  $\mathbf{H}$ . One popular method is by using protographs [7]. An LDPC code can be represented by using a bipartite Tanner graph consisting of  $n$  variable nodes (VNs) and  $c$  check nodes (CNs), where the  $i$ th CN is connected to the  $j$ th VN if  $h_{i,j} = 1$ . A protograph is also a bipartite graph defined by an adjacency matrix  $\mathbf{P} = [p_{i,j}] \in \mathbb{N}_0^{c' \times n'}$ , called the base matrix. Given  $\mathbf{P}$ , a parity-check matrix  $\mathbf{H}$  is obtained by replacing each entry  $p_{i,j}$  in  $\mathbf{P}$  with a random binary  $M$ -by- $M$  matrix which contains  $p_{i,j}$  ones in each row and column. This procedure is called lifting and  $M \geq \max_{i,j} p_{i,j}$  is the so-called lifting factor. Graphically, this construction amounts to copying the protograph  $M$  times and subsequently permuting edges. Parallel edges, i.e., for  $p_{i,j} > 1$ , are permitted in the protograph and are resolved in the lifting procedure. The design rate of the code is given by  $R = 1 - c/n = 1 - c'/n'$ , where  $c = c'M$  and  $n = n'M$ .



---

**Algorithm 1:** P-EXIT analysis of the WD for a  $(J, K)$  regular SC-LDPC protograph.

---

**Input:**  $l_{\max}$  (max. iterations per window),  $p_{\text{tar}}$  (target error probability),  $W$ ,  $(J', K')$ ,  $\rho$

**Output:**  $S$  (decoding success, either true or false),  $l_s$  (iterations until successful decoding)

```

1 for  $i = 1$  to  $n'$  do           /* initialization of channel variances for VNs */
2   if VN  $i$  is punctured set  $\sigma_i^2 = 0$            /* treat as an erasure */
3   else set  $\sigma_i^2 = f(\rho)$            /* E.g.,  $f(\rho) = 8R\rho$  if  $\rho = E_b/N_o$  [24] */
4  $l_s = 0$            /* total iteration counter */
5 for  $j = -W + 2$  to  $T$  do
6    $c_{\text{start}} \leftarrow \max((j - 1)J' + 1, 1)$            /* first index of active CNs */
7    $c_{\text{end}} \leftarrow \min((W + j - 1)J', m)$            /* last index of active CNs */
8    $v_{\text{start}} \leftarrow \max((j - 1)K' + 1, 1)$            /* first index of active VNs */
9    $v_{\text{end}} \leftarrow \min((W + j - 1)K', n')$            /* last index of active VNs */
10   $t_{\text{start}} \leftarrow \max((j - 1)K' + 1, 1)$            /* first index of target VNs */
11   $t_{\text{end}} \leftarrow \max((j - 1)K' + K', K')$            /* last index of target VNs */
12   $l = 0$ 
13  while  $l < l_{\max}$  do
14    if mean (error probability of VN  $t_{\text{start}}$  to  $t_{\text{end}}$ )  $< p_{\text{tar}}$  break while
15    for  $i = v_{\text{start}}$  to  $v_{\text{end}}$  compute Messages ( $\sigma_i^2$ ) of VN  $i$  /* Eq. (9.46) [24] */
16    for  $i = c_{\text{start}}$  to  $c_{\text{end}}$  compute Messages of CN  $i$  /* Eq. (9.47) [24] */
17     $l \leftarrow l + 1$  and  $l_s \leftarrow l_s + 1$ 
18 if mean (error probability of VN 1 to  $n'$ )  $< p_{\text{tar}}$  set  $S = 1$  else set  $S = 0$ 

```

---

of this tool for binary modulation is available in [19] and [24, Algorithm 9.2]. Here, we only describe the necessary modifications to account for the WD and the nonbinary modulations. We start with the former and explain the latter in the next section.

We employ the WD scheme developed in [10]. WD helps to alleviate the long decoding delays and high decoding complexity of SC-LDPC codes under full BP decoding by exploiting the fact that two VNs are not involved in the same parity-check equation if they are at least  $(m_s + 1)K'$  columns apart [10]. The WD restricts message updates to a subset of VNs and CNs in the entire graph. After a predetermined number of decoding iterations, this subset changes and the decoding window slides to the next position. Pseudocode for the modified P-EXIT analysis of SC-LDPC codes accounting for the WD is presented in Algorithm 1. The main difference with respect to BP decoding is the window size parameter  $W$ , which specifies the number of active CNs in the protograph considered in each window as a multiple of  $J'$ . The P-EXIT analysis for the standard BP decoder can be recovered from Algorithm 1 by setting  $T = 1$ ,  $W = 1$ ,  $J' = c'$ , and  $K' = n'$ .

## 4 Bit Mapper Optimization

### 4.1 Asymptotic Bit Mapper Model

Each VN in the protograph represents  $M$  VNs in the lifted Tanner graph. Since a VN corresponds to one bit in a codeword, the  $n'$  VNs in the protograph give rise to  $n'$  different classes of coded bits that are treated as statistically equivalent in the P-EXIT analysis. In particular, for binary modulation, each protograph VN is assigned with one input variance, corresponding to either a punctured bit or the Gaussian LLR channel (see lines 2 and 3 in Algorithm 1). For nonbinary modulations, VNs in the same class can in principle have different input densities. Assume for example that a given protograph is lifted with an even lifting factor  $M$  and coded bits are mapped consecutively to a 4-ary modulation. Then,  $M/2$  VNs in each class are allocated to the first modulation bit and  $M/2$  to the second.

We model the bit mapper by specifying the assignment of VN classes to the bit channels via a matrix  $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{m \times n'}$ , where  $a_{i,j}$ ,  $0 \leq a_{i,j} \leq 1 \forall i, j$  denotes the proportional allocation of VNs from the  $j$ th class (corresponding to the  $j$ th column in the base matrix) allocated to the  $i$ th bit in the signal constellation. The approaches in [15–17] can be recovered by considering only nonfractional assignments, i.e.,  $a_{i,j} \in \{0, 1\}$ . In that case, VNs of the original protograph [15, 16] or an intermediate protograph [17] are directly assigned to the modulation bits.

We point out that, instead of interpreting  $a_{i,j}$  as a deterministic fraction of VNs in a particular class allocated to a particular channel, one should interpret  $a_{i,j}$  as a probability, and study the bit mapper as a probabilistic mapping device that assigns coded bits randomly to channels, similar to [26]. Under this assumption, one may argue that the VNs belonging to a certain class “see” an equivalent bit channel which is the average of the original bit channels  $f_{L_{i,k}|B_{i,k}}(l|b)$ , weighted according to the probabilities  $a_{i,j}$ . The MI of each equivalent bit channel is a weighted average of the original channels’ MI as shown in the following lemma.

**Lemma 1.** *Let  $\{f_{L_i|B_i}(l|b) : 1 \leq i \leq m\}$  be a collection of symmetric LLR channels. Consider a new channel  $f_{L|B}(l|b)$ , where transmission takes place over the  $i$ th channel in the collection with probability  $\alpha_i$  and  $\sum_i \alpha_i = 1$ . Then  $I(L; B) = \sum_i \alpha_i I(L_i; B_i)$  for uniform input bits.*

*Proof.* The channel  $f_{L|B}(l|b)$  is a symmetric LLR channel. The claim then follows from  $f_{L|B}(l|b) = \sum_i \alpha_i f_{L_i|B_i}(l|b)$  and the fact that the MI between the output of a symmetric LLR channel  $f_{L|B}(l|b)$  and uniform input bits is  $I(L; B) = 1 - \int_{-\infty}^{\infty} f_{L|B}(l|0) \log_2(1 + e^{-l}) dl$ .  $\square$

If we collect the MI corresponding to the original  $m$  symmetric LLR channels in a vector  $\mathbf{I}(\rho) = (I_1(\rho), \dots, I_m(\rho))$ , then, multiplying  $\mathbf{I}(\rho)$  by  $\mathbf{A}$  leads to a vector  $(\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_{n'})$  with the MIs corresponding to the averaged bit channels as seen by the  $n'$  VN classes. These averaged bit channels are modeled as symmetric Gaussian LLR channels with

parameters  $(\sigma_1^2, \dots, \sigma_{n'}^2)$ . In particular, the P-EXIT analysis for nonbinary modulation is obtained by changing the initialization step in line 3 of Algorithm 1 and assigning  $\sigma_i^2 = J^{-1}(\tilde{I}_i)^2$ , where the algorithm takes  $\mathbf{A}$  as an additional input to compute  $\tilde{I}_i$  as described.

In order to have a valid probabilistic assignment, all columns in  $\mathbf{A}$  have to sum to one and all rows in  $\mathbf{A}$  have to sum to  $n'/m$ , i.e., we have  $mn'$  equality constraints in total. The first condition ensures that, asymptotically, all VNs are assigned to a channel, while the second condition ensures that all parallel channels are used equally often. The set of valid assignment matrices is denoted by  $\mathcal{A}^{m \times n'} \subset \mathbb{R}^{m \times n'}$ . In the case of punctured VNs, the corresponding columns in  $\mathbf{A}$  are removed and  $n'$  is interpreted as the number of *unpunctured* VNs.

## 4.2 Optimization

For a given bit mapper, i.e., for a given assignment matrix  $\mathbf{A}$ , an approximate decoding threshold  $\rho^*(\mathbf{A})$  can be found using Algorithm 1 as follows. Fix a certain precision  $\delta$ , target bit error probability  $p_{\text{tar}}$ , and maximum number of iterations  $l_{\text{max}}$ . Starting from some SNR  $\rho$  where Algorithm 1 converges to a successful decoding,  $S = 1$ , iteratively decrease  $\rho$  by  $\delta$  until the decoding fails. The smallest  $\rho$  for which  $S = 1$  is declared as the decoding threshold  $\rho^*(\mathbf{A})$ . For any  $\rho \geq \rho^*(\mathbf{A})$ , we denote the number of iterations until successful decoding by  $l_s(\mathbf{A}, \rho)$ .

We are interested in optimizing  $\mathbf{A}$  in terms of the decoding threshold for a given protograph and modulation format. The optimization problem is thus

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A} \in \mathcal{A}^{m \times n'}}{\text{argmin}} \quad \rho^*(\mathbf{A}), \quad (\text{A.6})$$

where the baseline system realizes a mapping of coded bits to modulation bits such that  $a_{i,j} = 1/m, \forall i, j$ , resulting in identical variances  $\sigma_i^2$  for the equivalent bit channels of all VN classes. The corresponding assignment matrix is denoted by  $\mathbf{A}_{\text{uni}}$ . The search space  $\mathcal{A}^{m \times n'}$  can be regarded as a convex polytope  $\mathcal{P}$  in  $p = (m-1)(n'-1)$  dimensions by removing the last row and column in  $\mathbf{A}$ , replacing the equality constraints with inequality constraints, and writing the matrix elements in a vector  $\mathbf{x} \in \mathbb{R}^p$  according to the prescription  $x_{(i-1)n'+j} = a_{i,j}$  for  $1 \leq i \leq m-1$  and  $1 \leq j \leq n'-1$ . While the search space is convex, one can show by simple examples that the objective function is nonconvex in  $\mathcal{P}$ . In the following, we discuss ways to obtain good bit mappers with reasonable effort. We also remark that some of the optimization approaches proposed previously in the context of bit mapper optimization for irregular LDPC codes are not necessarily appropriate in our case due to the higher number of VN classes, i.e., they can be too complex (for example the iterative grid search in [13]) or do not explore the search space efficiently (simple hill climbing approaches as in [14]).

First, as an alternative to directly optimizing the decoding threshold, we iteratively optimize the convergence behavior in terms of the number of iterations until successful

decoding as follows. Initialize  $\rho$  to the decoding threshold for the baseline bit mapper, i.e.,  $\rho = \rho^*(\mathbf{A}_{\text{uni}})$ . Find  $\mathbf{A}^*$  such that it minimizes the number of decoding iterations until convergence for the given  $\rho$ , i.e.,

$$\mathbf{A}^* = \underset{\mathbf{A} \in \mathcal{A}^{m \times n'}}{\operatorname{argmin}} \quad l_s(\mathbf{A}, \rho). \quad (\text{A.7})$$

For the found optimized  $\mathbf{A}^*$ , calculate the new decoding threshold  $\rho^*(\mathbf{A}^*)$ . If the threshold did not improve, stop. Otherwise, set  $\rho = \rho^*(\mathbf{A}^*)$  and repeat the optimization. The above iterative approach was already used by the authors to find good bit mappers for SC-LDPC codes in [20] for parallel BECs. This approach is largely based on the ideas presented in [27, Sec. IV], where optimized degree distributions for irregular LDPC codes are found. The computational complexity can be significantly reduced compared to the threshold minimization (A.6). However, it is not guaranteed to be equivalent to a true threshold optimization, i.e., in general  $\mathbf{A}_{\text{opt}} \neq \mathbf{A}^*$ . We employ differential evolution [28] to solve the optimization problem in (A.7), which has been previously applied by many authors in the context of irregular LDPC codes [24, p. 396]. Differential evolution is a solver for unconstrained optimization problems and we briefly indicate how the algorithm is modified to account for the constrained search space. First, since  $\mathcal{A}^{m \times n'}$  can be regarded as a convex polytope, it is straightforward to take uniformly distributed points for the initial population via standard random walk procedures [29]. Second, if the algorithm generates a trial point  $\mathbf{x}_t$  that lies outside the polytope, we apply the following randomized bounce-back strategy. Let  $\mathcal{L}$  be the line segment connecting  $\mathbf{x}_t$  and a random point inside the polytope, and let  $\mathbf{x}_i$  be the intersecting point of  $\mathcal{L}$  and the boundary of  $\mathcal{P}$ . We replace  $\mathbf{x}_t$  with a point taken randomly from  $\mathcal{L}$ , such that it lies in  $\mathcal{P}$  and has at most a distance  $d$  from  $\mathbf{x}_i$ , where  $d$  is the distance between  $\mathbf{x}_i$  and  $\mathbf{x}_t$ . For a detailed description of the algorithm itself and some guidelines regarding the optimization parameter choice, we refer the reader to [28].

The optimization complexity is further reduced by constraining the maximum number of iterations  $l_{\text{max}}$ . Practical systems commonly operate with a relatively small number of BP iterations. For example, in Sec. 5, we assume 50 BP iterations, and hence the decoding thresholds are optimized for the same number of iterations. In the simulative verification, we have observed that the performance of the finite-length codes assuming 50 BP iterations is generally better using a bit mapper that is also optimized for  $l_{\text{max}} = 50$  compared to, say,  $l_{\text{max}} = 1000$ , although the differences were small.

Additionally, for SC-LDPC codes, we take advantage of the structure of the optimized bit mappers for parallel BECs [20], which show a certain form of periodicity. The optimization complexity can then be reduced by assuming that the optimal solution lies in a lower-dimensional subspace of  $\mathcal{P}$ , defined by assignment matrices that take on a periodic form as  $\mathbf{A} = (\mathbf{A}', \mathbf{A}'', \mathbf{A}''', \dots, \mathbf{A}'', \mathbf{A}''')$ , with  $m \times V$  matrices  $\mathbf{A}'$ ,  $\mathbf{A}''$ , and  $\mathbf{A}'''$ , where  $V$  is the periodicity factor. If  $V$  is chosen small enough, the dimensionality of the search space (i.e.,  $(m-1)(3V-1)$ ) can be substantially reduced, which generally improves the convergence speed of the differential evolution algorithm.

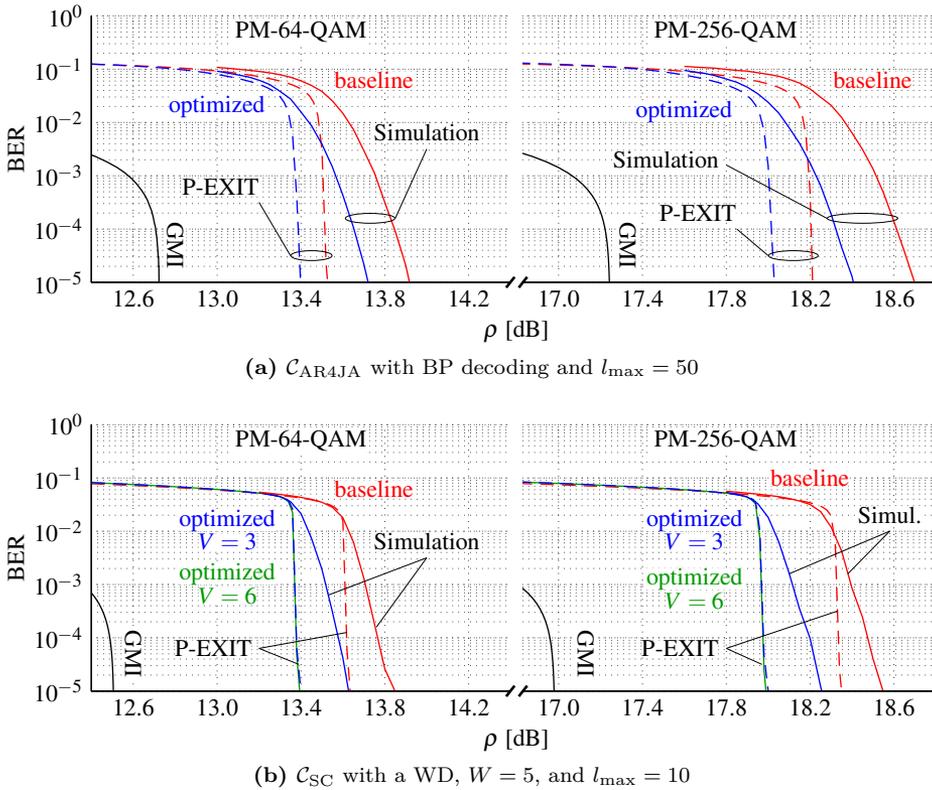
The methods and complexity reduction techniques described above have been selected to obtain a good trade-off between final performance and design complexity. In certain cases, for example the considered AR4JA code in the next section, it could be possible to further improve the performance at the expense of a higher design complexity by directly targeting the decoding threshold optimization (A.6) without the need for the iterative optimization (albeit we expect the improvements to be incremental). On the other hand, for the considered SC-LDPC code, the iterative optimization and periodicity assumptions were critical to maintain a reasonable design complexity, which is mainly due to the very large number of protograph VNs.

## 5 Results and Discussion

In this section, we present and discuss numerical results, and illustrate the performance gains that can be achieved by employing optimized bit mappers. For the baseline systems, we use a consecutive mapping of coded bits to modulation bits. Alternatively, one may use a uniformly random mapping, which has the same expected performance.

In order to show the flexibility of the technique, we consider four different scenarios, combining both modulation formats with one code based on the AR4JA protographs and one SC-LDPC code, where the lifting factor is  $M = 3000$  in all cases. For simplicity, the codes are randomly generated without further consideration of the graph structure. The protograph lifting procedure can in principle be combined with standard techniques to avoid short graph cycles that may potentially lead to high error floors [24, Ch. 6.3]. Alternatively, an additional outer algebraic code may be assumed, which removes remaining errors to achieve a required target BER of  $10^{-15}$ . A rate  $R = 2/3$  code based on the AR4JA protograph for  $\ell = 1$  is used, which is denoted by  $\mathcal{C}_{\text{AR4JA}}$ . For the spatially coupled case with  $T = 30$ , a code based on the protograph described in Sec. 3.2 is used, which is denoted by  $\mathcal{C}_{\text{SC}}$ . For the given value of  $T$ , the design rate is  $R(30) = 0.656$ . For the AR4JA code, standard BP decoding is assumed with  $l_{\text{max}} = 50$ , while for the SC-LDPC codes, we employ a WD with  $W = 5$  and  $l_{\text{max}} = 10$ , which again amounts to a total of 50 iterations per decoded bit. We also tried other combinations of  $W$  and  $l_{\text{max}}$  with a similar total number of iterations and this combination gave the best performance. For the bit mapper optimization and in particular the P-EXIT analysis, we use the same values for  $l_{\text{max}}$  and  $W$ , and additionally  $p_{\text{tar}} = 10^{-5}$ . The finite-length bit mappers are obtained via the rounded matrix  $M\mathbf{A}^*$  from which the index assignment of coded bits to modulation bits is determined.

Notice that in all four scenarios, the approaches in [15–17] are either not possible (due to a mismatch between the number of protograph VNs and the number of modulation bits) or not feasible (due to the large complexity of the resulting optimization). As an example, the protograph corresponding to  $\mathcal{C}_{\text{SC}}$  has 90 VNs and can be directly connected to the three distinct bit channels of PM-64-QAM. This leads, however, to a very large number of possible (nonfractional) connections between protograph VNs and modulation



**Figure 6:** Comparison of the optimized bit mappers (blue) with the baseline bit mappers (red) for the linear transmission scenario. Dashed lines correspond to P-EXIT analysis and solid lines to simulation results. In (b), solid green lines correspond to the P-EXIT analysis for  $V = 6$ .

bits.

## 5.1 Linear Transmission

We start by providing a verification of the proposed optimization technique assuming an AWGN channel. This case is obtained when nonlinear effects are ignored, i.e.,  $\gamma = 0$ . In this case, the channel PDF (A.2) is valid without approximations.

In Fig. 6(a), the predicted bit error rate (BER) of the AR4JA code via the P-EXIT analysis is shown together with Monte Carlo simulations by the dashed and solid lines, respectively. Performance curves for the baseline bit mappers are shown in red and for the optimized ones in blue. As a reference, we also plot the BER-constrained [24, p. 17] generalized mutual information (GMI) for the corresponding spectral efficiency in each figure (the GMI is also referred to as the BICM capacity [30]). For both scenarios,

it can be observed that the optimized bit mappers lead to a significant performance improvement. The gains that can be achieved at a BER of  $10^{-5}$  are approximately 0.19 and 0.25 dB for PM-64-QAM and PM-256-QAM, respectively. The predicted gains from the P-EXIT analysis for the same BER is slightly less, i.e., 0.12 and 0.19 dB, respectively. The deviation of the asymptotic analysis from the actual simulation results is to be expected due to the Gaussian approximation of the LLR densities and the finite lifting factor and, hence, finite block lengths of the codes. However, it is important to observe that, even though the optimization was carried out assuming a cycle-free graph structure, the predicted performance gains for the finite-length codes is well preserved.

Similarly, the performance of the SC-LDPC code is shown in Fig. 6(b). The periodicity factor for the bit mapper optimization was set to  $V = 3$ . The observed gains at a BER of  $10^{-5}$  are approximately 0.20 dB for PM-64-QAM and 0.25 dB for PM-256-QAM. We also show the predicted P-EXIT performance obtained for bit mappers that are optimized assuming a larger periodicity factor of  $V = 6$  by the solid green curves. It can be seen that for both modulation formats, the additional gains are incremental, i.e., for PM-64-QAM the predicted performance curves virtually overlap, while for PM-256-QAM, the difference is roughly 0.01 dB. This suggests that a full optimization of  $\mathbf{A}$  will be only marginally better than with  $V = 3$ .

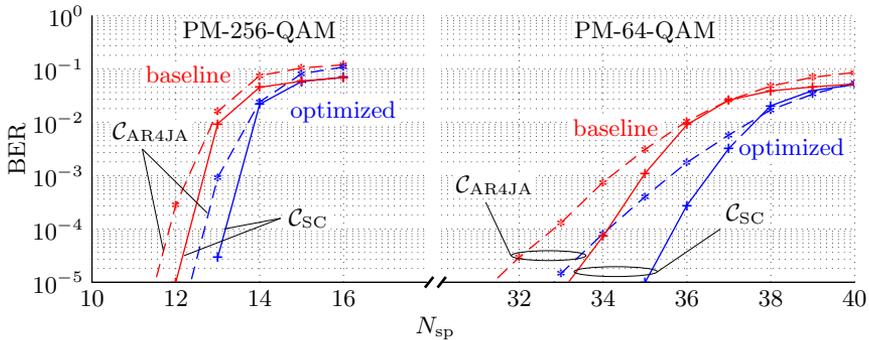
From Fig. 6, it appears that the P-EXIT analysis consistently underestimates the finite-length performance improvement for the AR4JA code, while it overestimates the improvement for the SC-LDPC code. This observation does, however, not apply in general and seems to be coincidental. In particular, we also optimized the bit mapper for AR4JA codes of different code rates (results not shown), and the P-EXIT analysis may also underestimate the true performance improvements in that case. Moreover, we would like to stress that a direct comparison between the two codes is difficult, because of the slightly different code rates (and hence spectral efficiencies) and different decoding complexities and delays. Fair comparisons between SC-LDPC codes and LDPC block codes is an active area of research and beyond the scope of this paper.

## 5.2 Nonlinear Transmission

In this section, we consider a transmission scenario including nonlinear effects, i.e.,  $\gamma \neq 0$ , where the assumed channel PDF (A.2) is only approximately valid. In particular, we study the potential increase in transmission reach that can be obtained by employing the optimized bit mappers.

We consider a single channel transmission scenario to keep the simulations within an acceptable time. In the simulation model, we assume perfect knowledge about the polarization state, and perfect timing and carrier synchronization. All chosen system parameters are summarized in Table 1. Additionally, we use a root-raised cosine pulse  $p(t)$  with a roll-off factor of 0.25. In order to solve (A.1), we employ the symmetric SSFM with two samples per symbol and a fixed step size of  $\Delta = (10^{-4}L_D^2L_{NL})^{1/3}$ , where  $L_D = 1/(|\beta_2|R_s^2)$  and  $L_{NL} = 1/(\gamma P)$  is the dispersive and nonlinear length, respectively.

parameter	meaning	value
$R_s$	symbol rate	40 Gbaud
$L_{\text{sp}}$	span length	70 km
$\alpha$	attenuation coefficient (0.25 dB/km)	$0.0576 \text{ km}^{-1}$
$\beta_2$	chromatic dispersion coefficient	$-21.668 \text{ ps}^2/\text{km}$
$\gamma$	nonlinear Kerr parameter	$1.4 \text{ W}^{-1} \text{ km}^{-1}$
$\nu_s$	carrier frequency (1550 nm)	$1.934 \times 10^{14} \text{ Hz}$
$n_{\text{sp}}$	spontaneous emission factor	1.622

**Table 1:** System parameters

**Figure 7:** Comparison of the optimized bit mappers (blue) with the baseline bit mappers (red) for the nonlinear transmission scenario.

The input power that maximizes  $\rho$  according to the GN model varies between  $-2.2$  dBm for  $N_{\text{sp}} = 10$  and  $-2.6$  dBm for  $N_{\text{sp}} = 40$ . For simplicity, the input power per polarization is fixed to  $P = -2.5$  dBm for all values of  $N_{\text{sp}}$ .

In Fig. 7, the simulated BER of the PM systems using  $\mathcal{C}_{\text{AR4JA}}$  and  $\mathcal{C}_{\text{SC}}$  is shown as a function of the number of fiber spans  $N_{\text{sp}}$  by the dashed and solid lines, respectively. Again, curves corresponding to the baseline bit mappers are shown in red, while curves corresponding to the optimized bit mappers are shown in blue. Notice that the SNR decrease (in dB) is not linear with increasing number of spans, hence the different slopes compared to the curves shown in Fig. 6. For PM-256-QAM, the transmission reach can be increased by roughly one additional span for both codes, at the expense of a slightly increased BER. For example, for  $\mathcal{C}_{\text{SC}}$ , the transmission reach can be increased from 12 to 13 spans, while the BER slightly increases from  $10^{-5}$  to  $3 \cdot 10^{-5}$ . For PM-64-QAM, the increase is roughly 1 span for  $\mathcal{C}_{\text{AR4JA}}$  and roughly 2 spans for  $\mathcal{C}_{\text{SC}}$ . In fact, these gains can be approximately predicted also from the GN model. For example, for the chosen input power and system parameters, the GN model predicts an SNR decrease of roughly 0.3 dB from  $N_{\text{sp}} = 12$  to  $N_{\text{sp}} = 13$  and 0.15 dB from  $N_{\text{sp}} = 34$  to  $N_{\text{sp}} = 35$ , i.e., one would expect the performance improvements in the linear transmission scenario to

translate into roughly one additional span for PM-256-QAM and one to two additional spans for PM-64-QAM. This estimate corresponds to an increase of the transmission reach by 3–8%, which is well in line with the simulation results presented in Fig. 7.

## 6 Conclusion

In this paper, we studied the bit mapper optimization for a PM fiber-optical system. Focusing on protograph-based codes, an optimization approach was proposed based on a fractional allocation of protograph bits to modulation bits via a modified P-EXIT analysis. Extensive numerical simulations were used to verify the analysis for a dispersion uncompensated link assuming both linear and nonlinear transmission regimes. The results show performance improvements of up to 0.25 dB, translating into a possible extension of the transmission reach by up to 8%.

## Acknowledgements

This work was partially funded by the Swedish Research Council under grant #2011-5961 and by the European Community's Seventh's Framework Programme (FP7/2007-2013) under grant agreement No. 271986. The simulations were performed in part on resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE.

## References

- [1] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, Feb. 2010.
- [2] D. J. Costello and G. D. Forney, Jr., "Channel coding: The road to channel capacity," *Proc. IEEE*, vol. 95, no. 6, pp. 1150–1177, Jun. 2007.
- [3] B. Smith and F. R. Kschischang, "A pragmatic coded modulation scheme for high-spectral-efficiency fiber-optic communications," *J. Lightw. Technol.*, vol. 30, no. 13, pp. 2047–2053, Jul. 2012.
- [4] I. B. Djordjevic, M. Arabaci, and L. L. Minkov, "Next generation FEC for high-capacity communication in optical transport networks," *J. Lightw. Technol.*, vol. 27, no. 16, pp. 3518–3530, Aug. 2009.
- [5] L. Beygi, E. Agrell, P. Johannisson, M. Karlsson, and H. Wymeersch, "A discrete-time model for uncompensated single-channel fiber-optical links," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3440–3450, Nov. 2012.

- [6] A. Carena, V. Curri, G. Bosco, P. Poggiolini, and F. Forghieri, "Modeling of the impact of nonlinear propagation effects in uncompensated optical coherent transmission links," *J. Lightw. Technol.*, vol. 30, no. 10, pp. 1524–1539, May 2012.
- [7] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progress Report 42-154*, JPL, 2005.
- [8] L. Schmalen, A. J. de Lind van Wijngaarden, and S. ten Brink, "Forward error correction in optical core and optical access networks," *Bell Labs Tech. J.*, vol. 18, no. 3, pp. 39–66, Mar. 2013.
- [9] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, St. Louis, Missouri, 2005.
- [10] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [11] A. J. Felström and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
- [12] S. Kudekar, T. Richardson, and R. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [13] T. Cheng, K. Peng, J. Song, and K. Yan, "EXIT-aided bit mapping design for LDPC coded modulation with APSK constellations," *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 777–780, Jun. 2012.
- [14] G. Richter, A. Hof, and M. Bossert, "On the mapping of low-density parity-check codes for bit-interleaved coded modulation," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Nice, Italy, 2007.
- [15] D. Divsalar and C. Jones, "Protograph based low error floor LDPC coded modulation," in *Proc. IEEE Military Communications Conf. (MILCOM)*, Atlantic City, NJ, 2005.
- [16] Y. Jin, M. Jiang, and C. Zhao, "Optimized variable degree matched mapping for protograph LDPC coded modulation with 16QAM," in *Proc. Int. Symp. Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, 2010.
- [17] T. Van Nguyen, A. Nosratinia, and D. Divsalar, "Threshold of protograph-based LDPC coded BICM for Rayleigh fading," in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Houston, TX, 2011.

- 
- [18] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [19] G. Liva and M. Chiani, “Protograph LDPC codes design based on EXIT analysis,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Washington, DC, 2007.
- [20] C. Häger, A. Graell i Amat, A. Alvarado, F. Brännström, and E. Agrell, “Optimized bit mappings for spatially coupled LDPC codes over parallel binary erasure channels,” in *Proc. IEEE Int. Conf. Communications (ICC)*, Sydney, Australia, 2014.
- [21] G. P. Agrawal, *Lightwave Technology: Telecommunication Systems*. Wiley-Interscience, 2005.
- [22] M. Secondini and E. Forestieri, “The nonlinear Schrödinger equation in fiber-optic systems,” *Riv. Mat. Univ. Parma*, vol. 8, pp. 69–98, Sep. 2008.
- [23] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, “Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003.
- [24] W. Ryan and S. Lin, *Channel Codes Classical and Modern*. Cambridge University, 2009.
- [25] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., “AWGN channel analysis of terminated LDPC convolutional codes,” in *Proc. Information Theory and Applications Workshop (ITA)*, La Jolla, CA, 2011.
- [26] R. Liu, P. Spasojevic, and E. Soljanin, “Reliable channel regions for good binary codes transmitted over parallel channels,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1405–1424, Apr. 2006.
- [27] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [28] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Opt.*, vol. 11, pp. 341–359, Nov. 1997.
- [29] D. Kaufman and R. Smith, “Direction choice for accelerated convergence in hit-and-run sampling,” *Operations Research*, vol. 46, no. 1, pp. 84–95, Jan. 1998.
- [30] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.



**Terminated and Tailbiting Spatially-Coupled Codes with Optimized Bit Mappings for Spectrally Efficient Fiber-Optical Systems**

Christian Häger, Alexandre Graell i Amat, Fredrik Brännström,  
Alex Alvarado, and Erik Agrell

*J. Lightw. Technol.* (invited paper), vol. 33, no. 7, pp. 1275–1285,  
April 2015

*The layout has been revised.*

## Abstract

We study the design of spectrally efficient fiber-optical communication systems based on different spatially-coupled (SC) forward error correction (FEC) schemes. In particular, we optimize the allocation of the coded bits from the FEC encoder to the modulation bits of the signal constellation. Two SC code classes are considered. The codes in the first class are protograph-based low-density parity-check (LDPC) codes which are decoded using iterative soft-decision decoding. The codes in the second class are generalized LDPC codes which are decoded using iterative hard-decision decoding. For both code classes, the bit allocation is optimized for the terminated and tailbiting SC cases based on a density evolution analysis. An optimized bit allocation can significantly improve the performance of tailbiting SC codes over the baseline sequential allocation, up to the point where they have a comparable gap to capacity as their terminated counterparts, at a lower FEC overhead. For the considered terminated SC codes, the optimization only results in marginal performance improvements, suggesting that in this case a sequential allocation is close to optimal.

## 1 Introduction

Designing spectrally efficient fiber-optical systems that can operate close to the capacity limits [1] has become an important research topic [2–4]. Such systems are often implemented according to the pragmatic bit-interleaved coded modulation (BICM) paradigm [5], where a single binary forward error correction (FEC) encoder is used in combination with a nonbinary signal constellation. A random allocation (or interleaving) [5] of the coded bits from the FEC encoder to the modulation bits of the signal constellation is commonly assumed. In this paper, we optimize the allocation to the modulation bits for a coherent long-haul polarization-multiplexed (PM) fiber-optical system. In particular, we consider different spatially-coupled (SC) FEC schemes both with soft-decision decoding (SDD) and hard-decision decoding (HDD).

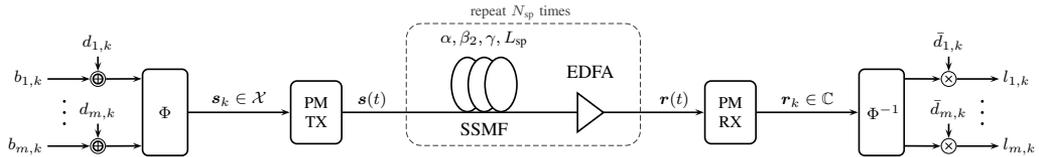
SC low-density parity-check (SC-LDPC) codes have attracted a great deal of attention in the recent years. They are considered as viable candidates for future spectrally efficient fiber-optical systems [3, 6, 7] due to their capacity-achieving performance for many communication channels [8]. SC-LDPC codes promise excellent belief propagation (BP) performance with a quasi-regular node degree distribution and low node degrees. The BP performance of SC-LDPC codes can further be improved by increasing the node degrees, whereas the decoding performance for regular LDPC codes generally worsens if the node degrees are increased [8]. While irregular LDPC codes can also perform close to capacity [9], the optimal degree distribution depends on the code rate and/or the channel [10].

High node degrees are also often required for good performance which leads to a high decoding complexity.

We consider two different SC code classes taken from the literature. The codes in the first class are protograph-based SC-LDPC codes [11,12] which are decoded using iterative SDD in the form of BP decoding. BP is a message passing algorithm in which “soft” (i.e., real-valued) messages are exchanged between the variable nodes (VNs) and check nodes (CNs) in the Tanner graph representing the code. The codes in the second class are SC generalized LDPC (SC-GLDPC) codes where each coded bit is protected by two  $t$ -error correcting Bose–Chaudhuri–Hocquenghem (BCH) component codes [13]. These codes are decoded using iterative HDD with bounded-distance decoding (BDD) of the component BCH codes. Iterative HDD can be seen as a message passing algorithm with “hard” (i.e., binary) messages in the Tanner graph representing the GLDPC code and is significantly less complex than SDD [14].

The adoption of SDD is considered one of the most important factors for increasing the performance of fiber-optical systems [15]. However, SDD poses implementation challenges at very high data rates motivating the use of less complex FEC schemes [14]. The SC-GLDPC codes we consider in this paper were proposed in [13], where it is shown that they can approach the capacity of the binary symmetric channel (BSC) under iterative HDD for high code rates (i.e., low FEC overheads (OHs)). We use these codes because a density evolution (DE) analysis is readily available in [13]. This allows us to apply the optimization techniques for protograph-based SC-LDPC codes we previously presented in [16] to the practically relevant case of SC-GLDPC codes with iterative HDD. The SC-GLDPC code ensemble in [13] is closely related to other recently proposed FEC schemes for optical transport networks, such as staircase codes [14] (which are themselves related to block-wise braided block codes [17]), and the modified construction of tightly-braided block codes proposed in [18]. For other related works on GLDPC codes for fiber-optical communications, we refer the interested reader to [19,20] and references therein.

The outstanding performance of SC codes is due to a termination boundary effect which initiates a wave-like decoding behavior [8]. This behavior of terminated SC codes comes at the price of a rate loss, i.e., a larger FEC OH, compared to the underlying uncoupled codes. So-called tailbiting SC codes provide an interesting solution to this problem, since they do not suffer from an increased OH. However, by default, a tailbiting SC code behaves essentially the same as the underlying uncoupled code due to the absence of a termination boundary. The main aim of this paper is to demonstrate that the unequal error protection offered by the modulation bits of a nonbinary signal constellation can be exploited to create an artificial termination boundary. This significantly improves the performance of tailbiting SC codes, both in the case of SDD and HDD. With an optimized bit allocation, the capacity gap of the considered tailbiting SC codes is comparable to the gap of their terminated counterparts, at a lower FEC OH. For the considered terminated SC codes, the performance gain due to an optimized bit allocation is limited, in particular for the SC-GLDPC codes with HDD. Simulation results for both linear and nonlinear



**Figure 1:** Block diagram of the considered PM transmission system.

transmission scenarios confirm the DE analysis.

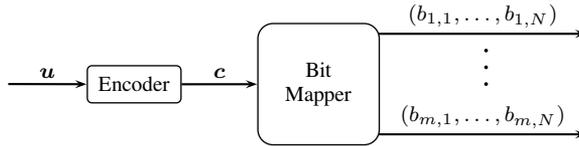
The remainder of the paper is organized as follows. In Section 2, the assumed PM transmission system is described. The two SC FEC schemes are covered in Sections 3 and 4, where we explain the code construction, the decoding algorithms, and the DE analysis with the help of several examples. In Section 5, we briefly review the optimization techniques for the bit allocation described in [16], which apply to the considered SC-LDPC codes with SDD. We also discuss how they are easily extended to the SC-GLDPC codes with HDD. Results are presented and discussed in Section 6 and the paper is concluded in Section 7.

## 2 System Model

A block diagram of the considered PM fiber-optical transmission system is shown in Fig. 1. At each discrete time instant  $k$ , the modulator  $\Phi$  takes  $m$  bits  $b_{i,k}$ ,  $i = 1, \dots, m$ , and maps them to a symbol  $\mathbf{s}_k = (s_{x,k}, s_{y,k})$  taken with uniform probabilities from a signal constellation  $\mathcal{X} \subset \mathbb{C}^2$  ( $|\mathcal{X}| = 2^m$ ) according to the binary labeling. The modulo-2 addition of the independent and uniformly distributed bits  $d_{i,k}$  (and the multiplication by  $\bar{d}_{i,k} = (-1)^{d_{i,k}}$  at the receiver) shown in Fig. 1 serves as a symmetrization technique [21].<sup>1</sup>

The baseband signal in polarization  $x$  is  $s_x(t) = \sum_k s_{x,k} p(t - k/R_s)$  with (real-valued) pulse shape  $p(t)$  and symbol rate  $R_s$  (and similarly for polarization  $y$ ). The transmit power  $P = \lim_{T \rightarrow \infty} \int_{-T}^T s_x(t)^2 dt / (2T)$  is assumed to be equal in both polarizations. The PM signal  $\mathbf{s}(t) = (s_x(t), s_y(t))$  is launched into the fiber and propagates according to the Manakov equation [23]. The optical link consists of  $N_{\text{sp}}$  spans of standard single-mode fiber (SSMF) with attenuation coefficient  $\alpha$ , group velocity dispersion  $\beta_2$ , nonlinear Kerr parameter  $\gamma$ , length  $L_{\text{sp}}$ , and a lumped amplification scheme (no optical dispersion compensation is assumed). Each erbium-doped fiber amplifier (EDFA) introduces circularly symmetric complex Gaussian noise with two-sided power spectral density  $N_{\text{EDFA}} = (e^{\alpha L_{\text{sp}}} - 1) h \nu_s n_{\text{sp}}$  per polarization [1], where  $h$  is Planck's constant,  $\nu_s$  the carrier frequency, and  $n_{\text{sp}}$  the spontaneous emission factor. A coherent linear receiver according to  $r_{x,k} = r_x(t) \otimes h(t) \otimes p(-t)|_{t=k/R_s}$  is used in each polarization, where  $\otimes$  denotes convolution and  $h(t)$  is the impulse response of an equalizer which accounts

<sup>1</sup>The symmetrization makes the bit error probability independent of the transmitted bits. This is an important requirement for the all-zero codeword assumption which is commonly made in DE [22, p. 389].



**Figure 2:** Illustration of the Bit Mapper.

for linear distortions due to chromatic dispersion. The frequency response of the equalizer is given by  $H(f) = \exp(j2\beta_2\pi^2 f^2 N_{\text{sp}}L_{\text{sp}})$ . The received symbols are denoted by  $\mathbf{r}_k = (r_{x,k}, r_{y,k})$ .

Two different demodulators  $\Phi^{-1}$  are considered. For SDD, the demodulator computes “soft” reliability information about the bits  $b_{i,k}$  in the form of log-likelihood ratios (LLRs)  $l_{i,k}$ . For HDD, the demodulator performs a minimum distance symbol-by-symbol detection of the received symbols with respect to the signal constellation  $\mathcal{X}$  and outputs the binary labeling associated with the detected symbol. Both demodulators are based on the assumption that the discrete-time channel from  $\mathbf{s}_k$  to  $\mathbf{r}_k$  is the additive white Gaussian noise (AWGN) channel with signal-to-noise ratio (SNR) denoted by  $\rho$ . This assumption is accurate for linear transmission (i.e.,  $\gamma = 0$ ) where  $\rho = P/(N_{\text{sp}}N_{\text{EDFA}}R_s)$ . For the considered setup without optical inline dispersion compensation, it has been shown that this assumption is also justified, provided that dispersive effects are dominant and nonlinear effects are weak [24, 25]. For this case,  $\rho$  can be computed using [24, eq. (15)] assuming single channel transmission. Under the Gaussian Noise model assumption, see [25] and references therein, similar expressions for the SNR are also computable for wavelength-division multiplexing systems.

We consider a system according to the BICM paradigm, where a binary code  $\mathcal{C} \subset \{0, 1\}^{n_c}$  of length  $n_c$  and dimension  $k_c$  is employed and each codeword  $\mathbf{c} = (c_1, \dots, c_{n_c})$  is transmitted using  $N = n_c/m$  symbols  $\mathbf{s}_k$ . The allocation of the coded bits to the modulation bits is determined by a bit mapper<sup>2</sup> as shown in Fig. 2, where  $\mathbf{u} = (u_1, \dots, u_{k_c})$  is the information word. The bit mapper optimization is discussed in Section 5. The optimization is based on the AWGN channel model because a direct optimization using DE for the optical channel defined by the nonlinear Schrödinger equation is not feasible. The accuracy of this approach is verified through simulation results for a nonlinear transmission scenario in Section 6.4. To the best of our knowledge, there are no comparable works by other authors on bit mapper optimization for SC codes. Hence, as a baseline for a comparison, we use a sequential mapper according to  $b_{i,k} = c_{(k-1)m+i}$  for  $1 \leq i \leq m$ ,  $1 \leq k \leq N$ . For the considered codes, a sequential mapper has the same expected performance as a random mapper.

<sup>2</sup>The bit mapper should not be confused with the modulator  $\Phi$ , which is sometimes also referred to as a mapper. In the literature, the term “bit interleaver” is also frequently used instead of “bit mapper”.

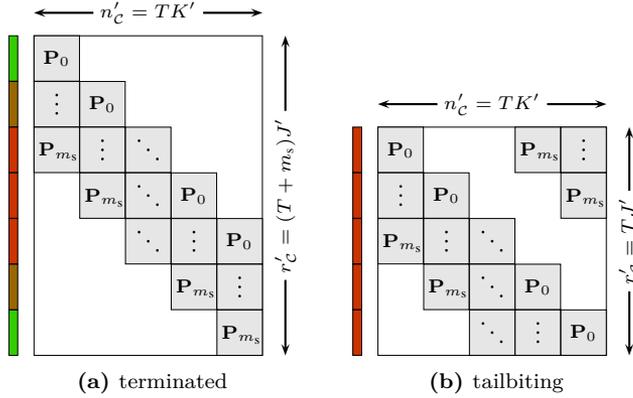


Figure 3: Base matrices  $\mathbf{P}$  for protograph-based SC-LDPC codes.

## 3 Protograph-Based SC-LDPC Codes

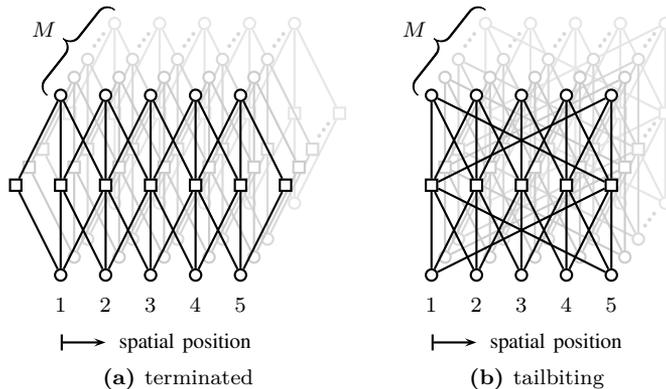
### 3.1 Code Construction

An LDPC code of length  $n_c$  and dimension  $k_c$  is defined via a sparse parity-check matrix  $\mathbf{H} = [h_{i,j}] \in \{0, 1\}^{r_c \times n_c}$ , where  $r_c \geq n_c - k_c$  with equality if and only if  $\mathbf{H}$  has full rank. One popular method to construct LDPC codes is by using protographs [11]. A protograph is a bipartite graph defined by an adjacency matrix  $\mathbf{P} = [p_{i,j}] \in \mathbb{N}_0^{r'_c \times n'_c}$ , called the base matrix, where  $\mathbb{N}_0$  is the set of nonnegative integers. Given  $\mathbf{P}$ , the parity-check matrix  $\mathbf{H}$  is obtained by replacing each entry  $p_{i,j}$  in  $\mathbf{P}$  with a random binary  $M$ -by- $M$  matrix which contains  $p_{i,j}$  ones in each row and column. This procedure is called lifting and  $M \geq \max_{i,j} p_{i,j}$  is the so-called lifting factor. The design rate of the code is given by  $R = 1 - r_c/n_c = 1 - r'_c/n'_c$ , where  $r_c = r'_c M$  and  $n_c = n'_c M$ .

SC-LDPC codes have parity-check matrices with a band-diagonal structure and can be constructed using protographs [12]. The base matrix of a  $(J, K)$  regular SC-LDPC code with spatial length  $T$  is constructed by specifying matrices  $\mathbf{P}_i$ ,  $0 \leq i \leq m_s$ , of dimension  $J'$  by  $K'$ , where  $m_s$  is referred to as the memory. The matrices are such that  $\sum_{i=0}^{m_s} \mathbf{P}_i$  has column weight  $J$  and row weight  $K$  for all columns and rows. Given the matrices  $\mathbf{P}_i$  and the spatial length  $T$ , one can construct  $\mathbf{P}$  as shown in Fig. 3(a) for the terminated case and in Fig. 3(b) for the tailbiting case.<sup>3</sup> Terminated and tailbiting SC-LDPC codes have design rates  $R(T) = 1 - J'/K' - m_s J'/(TK')$  and  $R = 1 - J'/K'$ , respectively [12]. The rate loss for the terminated code with respect to the tailbiting (or the underlying uncoupled regular) code can be made arbitrary small by letting  $T \rightarrow \infty$ , but this also leads to very long block lengths  $n_c = TK'M$  (assuming a fixed lifting factor  $M$ ).

*Example 1.* Consider the (3, 6) regular SC-LDPC code with  $\mathbf{P}_0 = \mathbf{P}_1 = \mathbf{P}_2 = (1, 1)$ ,

<sup>3</sup>The terminology originates from the trellis representation of convolutional codes, where the initial and final states are either determined by known bits (terminated) or forced to be identical (tailbiting).



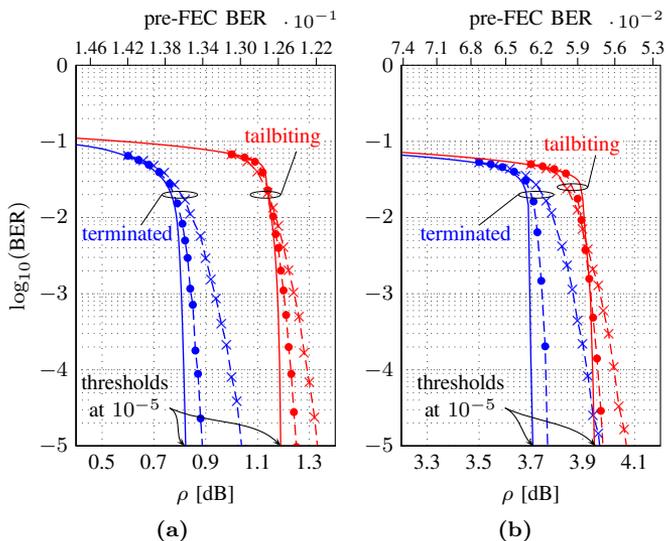
**Figure 4:** Protographs for the SC-LDPC code with  $T = 5$  in Example 1. The first step of the lifting procedure to obtain the Tanner graph (i.e., copying the protograph  $M$  times) is indicated in light gray.

$T = 5$ ,  $J' = 1$ ,  $K' = 2$ , and  $m_s = 2$ . The two protographs corresponding to the terminated and tailbiting cases are shown in Fig. 4. The design rates are  $R(5) = 0.3$  and  $R = 0.5$ , respectively.  $\triangle$

### 3.2 Soft-Decision Decoding and Density Evolution

The protograph-based SC-LDPC codes are decoded using the standard BP decoding [22, Sec. 5.4]. In order to alleviate the long decoding delay and high decoding complexity of SC-LDPC codes under full BP decoding, we employ the windowed decoder (WD) with a window size  $W$  developed in [26]. The WD reduces the decoding delay for terminated SC-LDPC codes from  $TMK'$  to  $WMK'$  coded bits [26]. For tailbiting SC-LDPC codes, additional memory for  $(m_s + W - 1)MK'$  values is required compared to terminated SC-LDPC codes, in order to take the circular wrap-around of the parity-check matrix into account. In particular, assume that the decoding starts when the channel observations corresponding to spatial positions 1 to  $m_s + W$  are received and the first targeted symbols are at position  $m_s + 1$ . (Due to the circular structure, the last targeted symbols are at position  $m_s$ .) Then, the observations corresponding to the first  $m_s$  positions as well as the final LLRs for the bits at positions  $m_s + 1$  to  $m_s + W - 1$  have to be stored. We also point the interested reader to [27], where the decoding of tailbiting SC-LDPC codes based on a pipeline decoder architecture is discussed.

The main tool for the analysis of LDPC codes under BP decoding is DE [28]. DE mimics the decoding process under a cycle-free graph assumption by tracking how the densities of the LLRs evolve with iterations. Tracking the full densities (or quantized densities in practice) is computationally demanding and extrinsic information transfer



**Figure 5:** Predicted (solid lines) and finite-length (dashed lines) performance for the codes in Examples 2 (left) and 6 (right). The codes have lengths 80 000 (a, crosses), 400 000 (a, dots), 168 000 (b, crosses), and 1 260 000 (b, dots).

(EXIT) functions [29] are usually considered to be a good compromise between computational efficiency and accuracy. For the protograph-based codes, we employ a modified protograph EXIT (P-EXIT) analysis [30] which accounts for the different protection levels of a nonbinary signal constellation and the WD, see [16, Algorithm 1].

*Example 2.* Consider the (3, 6) regular SC-LDPC code with  $\mathbf{P}_0 = (2, 2)$ ,  $\mathbf{P}_1 = (1, 1)$ ,  $T = 20$ ,  $J' = 1$ ,  $K' = 2$ , and  $m_s = 1$ , with rates  $R(T) = 0.475$  and  $R = 0.5$ , respectively. This is a slightly different construction compared to the one in Example 1 and the resulting codes are better suited for the use of a WD, see [26, Design Rule 1]. Assume that transmission takes place using PM-QPSK in the linear regime and a WD with  $W = 10$  and  $l_{\max} = 7$  is used. In Fig. 5(a), we show the predicted bit error rate (BER) obtained via the P-EXIT analysis (solid lines) together with the actual performance of randomly generated codes (dashed lines) for two different lifting factors  $M = 2000$  (crosses) and  $M = 10000$  (dots) for both the terminated (blue) and tailbiting (red) cases. Due to graph cycles, there is a mismatch between the actual performance and the DE prediction, in particular for the smaller lifting factor. However, the P-EXIT analysis accurately predicts the SNR region where the finite-length BER curves “bend” into their characteristic waterfall behavior.  $\triangle$

In Fig. 5(a), we also indicate the two points where the P-EXIT performance curves cross a BER of  $10^{-5}$ . We refer to the SNR value of such a point as the decoding threshold  $\rho^*$  for a target BER =  $10^{-5}$  and a given finite number of decoding iterations.

The thresholds can be numerically computed using a bisection search over a given SNR range. The thresholds are given by  $\rho^* \approx 0.82$  dB and  $\rho^* \approx 1.19$  dB for the terminated and tailbiting codes, respectively. The better decoding thresholds and finite-length performance of the terminated code can be explained by inspecting the structure of the base matrix Fig. 3(a). One may verify that the CN degrees corresponding to the first and last couple of rows are lower than the CN degrees corresponding to the rows in between (see also Fig. 4(a)). The lower degree CNs lead to a locally better decoding capability, which is visualized by the colored scale (green indicates a better correction capability), at the expense of a rate loss. This termination boundary effect initiates the wave-like decoding behavior that is characteristic for terminated SC-LDPC codes [8]. On the other hand, for the tailbiting case, all CNs have the same degree  $J$ , hence no rate loss is incurred and all positions are protected equally. However, this also prevents the initiation of a decoding wave.

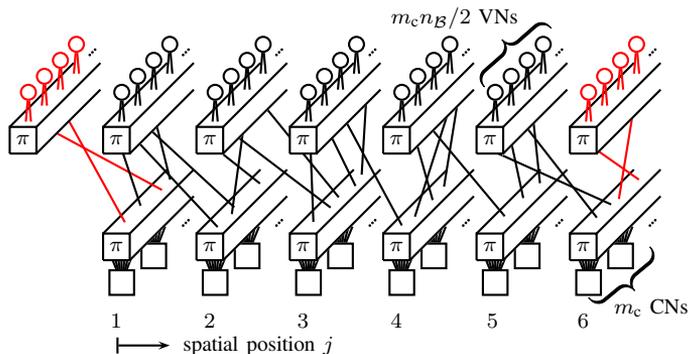
## 4 SC-GLDPC Codes with BCH Component Codes

### 4.1 Code Construction

We consider the  $(\mathcal{B}, m_c, T, w)$  SC-GLDPC code ensemble proposed in [13], where  $\mathcal{B}$  is a binary linear code of length  $n_{\mathcal{B}}$  and dimension  $k_{\mathcal{B}}$  that can correct all error patterns of weight at most  $t$ ,  $m_c$  is the number of CNs per spatial position,  $T$  is the spatial length, and  $w$  is the coupling size. In the following, we assume that  $\mathcal{B}$  is a shortened primitive BCH code with parameters  $(n_{\mathcal{B}}, k_{\mathcal{B}}) = (2^\nu - 1 - s, 2^\nu - \nu t - 1 - s)$ , where  $\nu$  is the Galois field extension degree and  $s$  is the number of shortened information bits. The code  $\mathcal{B}$  defines the constraints that have to be satisfied by all CNs in the Tanner graph representing the SC-GLDPC code.

For completeness, we review the construction of the terminated case described in [13, Def. 2] and explain the necessary modifications for the tailbiting case. Assume that  $m_c$  CNs with degree  $n_{\mathcal{B}}$  are placed at each of the spatial positions 1 to  $T + w - 1$  and  $m_c n_{\mathcal{B}}/2$  VNs of degree 2 are placed at each of the spatial positions 1 to  $T$ . Additionally,  $m_c n_{\mathcal{B}}/2$  VNs initialized to a known value are placed at positions  $j < 1$  and  $j > T$  to terminate the code. The connections between CNs and VNs are as follows. The  $m_c n_{\mathcal{B}}$  VN and CN sockets at each position are partitioned into  $w$  groups of equal size  $m_c n_{\mathcal{B}}/w$  via a uniform random permutation. The  $i$ -th group at the  $j$ -th VN position and the  $i$ -th group at the  $j$ -th CN position are denoted by  $\mathcal{S}_{j,i}^{(v)}$  and  $\mathcal{S}_{j,i}^{(c)}$ , respectively, where  $i \in \{0, 1, \dots, w - 1\}$ . The Tanner graph of one particular code in the ensemble is constructed by using a uniform random permutation to connect  $\mathcal{S}_{j,i}^{(v)}$  to  $\mathcal{S}_{j+i, w-i-1}^{(c)}$  and mapping the  $m_c n_{\mathcal{B}}/w$  edges between the two groups. For the tailbiting case, the position index  $j + i$  is interpreted modulo  $T$  and no known VNs are present.

*Example 3.* Consider the case where  $T = 5$  and  $w = 2$ . The Tanner graph of a code in the terminated ensemble is shown in Fig. 6. The blocks  $\pi$  spread out the edges from the



**Figure 6:** Tanner graph for a code in the terminated SC-GLDPC ensemble with  $T = 5$  and  $w = 2$ . Known VNs are shown in red.

VNs and CNs according to the random permutations in the construction. A code from the tailbiting ensemble is obtained by removing the known VNs and the CNs at position 6, and connecting the loose edges to the CNs at position 1.  $\triangle$

The design rate for the terminated ensemble is lower bounded by [31, eq. (2.2)]

$$R'(T) \geq R' - (1 - R') \frac{w - 1}{T}, \quad (\text{B.1})$$

where  $R' = 2k_{\mathcal{B}}/n_{\mathcal{B}} - 1$  is the design rate for the tailbiting ensemble. An exact expression for the design rate can be obtained by explicitly considering the possibility that certain CNs are connected exclusively to known VNs, similar to [8, Lemma 3]. However, for the high CN degrees and small coupling factors considered in this paper, one can safely ignore this possibility and we henceforth interpret (B.1) as an equality.

*Example 4.* Let  $\mathcal{B}$  be a shortened BCH code with  $\nu = 7$ ,  $t = 3$ , and  $s = 43$ , i.e.,  $\mathcal{B}$  has rate 0.75. For the terminated and tailbiting ensembles in Example 3, the design rates are given by  $R'(T) = 0.4$  and  $R' = 0.5$ , respectively.  $\triangle$

Similar to the parity-check matrix of an LDPC code, a GLDPC code can be specified by an incidence matrix [22, p. 220]. The dimensions of the incidence matrix are  $m_c(T + w - 1) \times Tm_c n_{\mathcal{B}}/2$  and  $m_c T \times Tm_c n_{\mathcal{B}}/2$  for the terminated and tailbiting ensembles, respectively.

*Example 5.* Consider the case where  $w = 2$  and  $T = \infty$ . Let  $n_{\mathcal{B}}$  be even and  $m_c = n_{\mathcal{B}}/2$ . If the edge permutations are such that the semi-infinite incidence matrix is the one shown in [32, p. 54], the code corresponds to a staircase code. In other words, staircase codes are contained in the terminated ensemble for a certain choice of parameters  $w$ ,  $T$ , and  $m_c$ .  $\triangle$

## 4.2 Hard-Decision Decoding and Density Evolution

We use the iterative HDD algorithm based on extrinsic message passing of binary messages that is proposed in [13, Sec. II-A] (see also [18, Sec. II-C]). Assume transmission over a BSC with crossover probability  $p$ . All outgoing VN messages are initialized to the channel observation. For each CN, the incoming messages from the VNs are collected in a candidate decoding vector, which is then decoded using BDD. The outgoing CN messages are computed based on the Hamming distance between the candidate vector and the decoded vector, cf. [18, Algorithm 1]. In the next iteration, the outgoing VN message on a particular edge corresponds to the incoming message on the other edge of that VN. Decoding continues for  $l_{\max}$  iterations. The final decision for each VN is made based on the channel observation and the two incoming messages. If the two messages agree, the bit is set to the message value. If the messages disagree, the bit is set to the binary complement of the channel observation. As pointed out in [13], extrinsic message passing is different compared to the conventional approach of decoding product-like codes (referred to as intrinsic message passing in [13]) and can be rigorously analyzed via DE even in the event of miscorrection [13], i.e., when undetected errors remain after BDD.

We briefly summarize the DE analysis presented in [13]. Assume that the all-zero codeword is transmitted and let  $q_j^{(l)}$  be the average probability that a message emitted by a VN at position  $j$  is in error (i.e., the message is “1”) after the  $l$ th iteration. The DE recursion is given by [13, eq. (5)]

$$q_j^{(l)} = \frac{1}{w} \sum_{k=0}^{w-1} f_{n_{\mathcal{B}}} \left( \frac{1}{w} \sum_{k'=0}^{w-1} q_{j-k'+k}^{(l-1)}; p \right), \quad (\text{B.2})$$

with [13, eq. (2)]

$$f_{n_{\mathcal{B}}}(x; p) \triangleq \sum_{i=0}^{n_{\mathcal{B}}-1} \binom{n_{\mathcal{B}}-1}{i} x^i (1-x)^{n_{\mathcal{B}}-1-i} \cdot (pP_{n_{\mathcal{B}}}(i) + (1-p)Q_{n_{\mathcal{B}}}(i)), \quad (\text{B.3})$$

where  $P_{n_{\mathcal{B}}}(i)$  and  $Q_{n_{\mathcal{B}}}(i)$  are defined in [13, eq. (3)] and [13, eq. (4)]. The initial conditions are  $q_j^{(0)} = p$  for  $j \in \{1, \dots, T\}$  and  $q_j^{(l)} = 0$  for  $j \notin \{1, \dots, T\}$ . For tailbiting ensembles, the subscript  $j - k' + k$  in (B.2) is calculated modulo  $T$ .

The analysis in [13] is presented for unshortened BCH codes, i.e.,  $s = 0$ . However, CNs connected to known variable nodes are treated as shortened component codes by adjusting the effective error probability of the incoming messages through the boundary condition  $q_j^{(l)} = 0$  for  $j \notin \{1, \dots, T\}$ . If the component codes are shortened BCH codes, one can therefore apply the same analysis as before, where  $n_{\mathcal{B}}$  now denotes the length of the unshortened code and the function  $f_{n_{\mathcal{B}}}(x; p)$  is replaced by  $f_{n_{\mathcal{B}}}(x(n_{\mathcal{B}} - s)/n_{\mathcal{B}}; p)$ .

The BER for the VNs at position  $j$  after the  $l$ th iteration was not derived in [13], but can be easily found as follows. First, we rewrite (B.3) in the form

$$f_{n_{\mathcal{B}}}(x; p) = p f_{n_{\mathcal{B}}}^{1 \rightarrow 1}(x) + (1-p) f_{n_{\mathcal{B}}}^{0 \rightarrow 1}(x), \quad (\text{B.4})$$

where  $f_{n_B}^{1 \rightarrow 1}(x)$  and  $f_{n_B}^{0 \rightarrow 1}(x)$  are implicitly defined via (B.3). We introduce the two variables

$$a_j^{(l)} \triangleq f_{n_B}^{1 \rightarrow 1} \left( \frac{1}{w} \sum_{k'=0}^{w-1} q_{j-k'}^{(l)} \right), \quad b_j^{(l)} \triangleq f_{n_B}^{0 \rightarrow 1} \left( \frac{1}{w} \sum_{k'=0}^{w-1} q_{j-k'}^{(l)} \right) \quad (\text{B.5})$$

and their averages

$$\bar{a}_j^{(l)} \triangleq \frac{1}{w} \sum_{k=0}^{w-1} a_{j+k}^{(l)} \quad \text{and} \quad \bar{b}_j^{(l)} \triangleq \frac{1}{w} \sum_{k=0}^{w-1} b_{j+k}^{(l)}. \quad (\text{B.6})$$

With these definitions, the recursion (B.2) becomes

$$q_j^{(l)} = p \bar{a}_j^{(l-1)} + (1-p) \bar{b}_j^{(l-1)} \quad (\text{B.7})$$

and the decoding error probability after the  $l$ th iteration is

$$p_{e,j}^{(l)} = p \left( \bar{a}_j^{(l-1)} \right)^2 + (1-p) \left( 1 - \left( 1 - \bar{b}_j^{(l-1)} \right)^2 \right). \quad (\text{B.8})$$

The final BER after  $l_{\max}$  steps of iterative HDD is computed as  $p_e = \frac{1}{T} \sum_{j=1}^T p_{e,j}^{(l_{\max})}$ .

Since we intend to use the DE analysis in an optimization routine, we approximate the two functions  $f_{n_B}^{1 \rightarrow 1}(x)$  and  $f_{n_B}^{0 \rightarrow 1}(x)$  with their high-rate scaling limit versions (i.e., for  $n_B \rightarrow \infty$ ) which are easier to compute and given by [13]

$$f_{n_B}^{1 \rightarrow 1}(x) \approx \phi(n_B x; t-1) \quad (\text{B.9})$$

and

$$f_{n_B}^{0 \rightarrow 1}(x) \approx \frac{1}{n_B(t-1)!} \phi(n_B x; t), \quad (\text{B.10})$$

where  $\phi(\lambda; t) = 1 - \sum_{i=0}^t \frac{\lambda^i}{i!} e^{-\lambda}$ .

It is straightforward to modify the decoding algorithm and the DE analysis if a similar WD as for the protograph-based SC-LDPC codes is used and hence we omit the details.

*Example 6.* Consider the case where  $T = 20$  and  $w = 2$ . Let  $\mathcal{B}$  be the same BCH code as in Example 3. The design rates are  $R'(T) = 0.475$  and  $R' = 0.5$ , respectively. Assume transmission using PM-QPSK in the linear regime and a WD with  $W = 5$  and  $l_{\max} = 10$ . In Fig. 5(b), we show the predicted BER obtained via DE (solid lines) together with the actual performance of randomly generated codes (dashed lines) for  $m_c = 200$  (crosses) and  $m_c = 1500$  (dots) for both the terminated (blue) and tailbiting (red) cases. The decoding thresholds at a BER of  $10^{-5}$  are  $\rho^* \approx 3.71$  dB and  $\rho^* \approx 3.94$  dB, respectively.  $\triangle$

## 5 Bit Mapper Optimization

The different modulation bits of a nonbinary signal constellation have different protection levels, which can be taken advantage of by optimizing the bit mapper. This concept is easiest to understand for HDD, which we describe first.

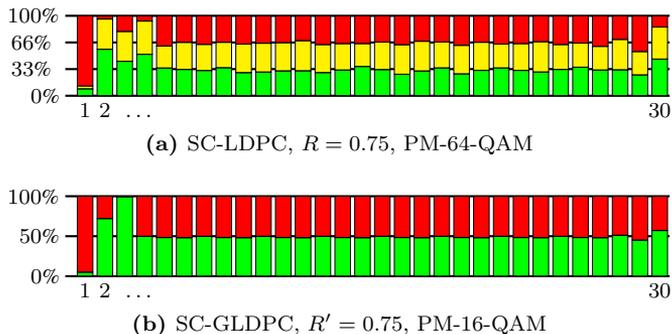
For the “hard” demodulator, the entire block diagram shown in Fig. 1 can be replaced by  $m$  parallel BSCs with different crossover probabilities  $p_i$ ,  $1 \leq i \leq m$ , which depend on the signal constellation, binary labeling, and SNR  $\rho$ . Each VN corresponds to a coded bit, and for the SC-GLDPC codes there are  $m_c n_B / 2$  VNs at each spatial position (see Section 4.1). The baseline bit mapper (see Section 2) allocates the same number of coded bits from each spatial position to the different modulation bits (i.e., the  $m$  parallel BSCs). In this case, the crossover probability for the VNs at an arbitrary spatial position is simply the average  $\bar{p} = \frac{1}{m} \sum_{i=1}^m p_i$ . More generally, the bit mapper is modeled by specifying the assignment of VNs to the modulation bits via a matrix  $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{m \times T}$ , where  $a_{i,j}$ ,  $0 \leq a_{i,j} \leq 1 \forall i, j$ , denotes the proportional allocation of the coded bits corresponding to the VNs at spatial position  $j$  allocated to the  $i$ th modulation bit, and  $\sum_{i=1}^m a_{i,j} = 1$ , for all  $j$ . The effective crossover probability for the VNs at spatial position  $j$  is therefore a weighted average of the BSC crossover probabilities according to  $\varepsilon_j = \sum_{i=1}^m a_{i,j} p_i$ . To account for different crossover probabilities at the spatial positions in the DE analysis, we can simply replace  $p$  in (B.2) by  $\varepsilon_j$ .

For the protograph-based SC-LDPC codes with SDD, one can make similar considerations. Each VN in the protograph represents  $M$  VNs in the lifted Tanner graph, i.e.,  $M$  coded bits. If we assume for example that a given protograph is lifted with a lifting factor  $M$  which is divisible by  $m$ , the baseline bit mapper allocates  $M/m$  coded bits for each protograph VN to each modulation bit. The bit mapper is modeled via a matrix  $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{m \times n'_c}$ , where  $a_{i,j}$  now denotes the proportional allocation of the coded bits corresponding to the  $j$ th column in the base matrix allocated to the  $i$ th modulation bit. The matrix  $\mathbf{A}$  is then used in the modified P-EXIT analysis to predict the iterative performance behavior under SDD [16, Algorithm 1].

We optimize  $\mathbf{A}$  based on the decoding threshold with the help of differential evolution [33]. For more details about the optimization procedure, we refer the reader to [16], where we also discuss several techniques to reduce the optimization complexity for SC codes. Once an optimized bit mapping matrix  $\mathbf{A}^*$  is found, the finite-length bit mapper is obtained via the rounded matrix  $(m_c n_B / 2) \mathbf{A}^*$  for the SC-GLDPC codes and via  $M \mathbf{A}^*$  for the SC-LDPC codes, from which the index assignment of coded bits to modulation bits is determined.

## 6 Results and Discussion

Since this paper does not deal with code design, we rely on code parameters that have been proposed elsewhere in the literature in order to illustrate the bit mapper optimiza-



**Figure 7:** Optimized allocation to the modulation bits with the best (green), worst (red), and intermediate (yellow, for PM-64-QAM) protection level for each spatial position of the tailbiting code in two scenarios.

tion technique. For the numerical results, we consider protograph-based SC-LDPC codes with  $\mathbf{P}_0 = (1, 2, 1, 2)$  and  $\mathbf{P}_1 = (3, 2, 3, 2)$ , where  $J' = 1$ ,  $K' = 4$ , and  $m_s = 1$  [34]. The design rate of the tailbiting case is  $R = 0.75$  (OH = 33%). For the SC-GLDPC codes, we restrict ourselves to  $w = 2$  and use the BCH code parameters in [35, Table I] which are optimized for staircase codes. In particular, we consider  $\nu = 9$ ,  $t = 4$ , and  $s = 223$ , which again leads to  $R' = 0.75$ . The staircase code for these parameters is estimated to perform approximately 1.38 dB away from the BSC capacity (at a BER of  $10^{-15}$ ) [35, Table I]. We also consider an example with higher rate and performance closer to capacity. In particular, we consider  $\nu = 10$ ,  $t = 4$ ,  $s = 143$  where  $R' = 0.91$  (OH = 10%). The gap to capacity of the staircase code for these parameters is estimated to be 0.59 dB [35, Table I].

The bit mapper optimization is performed for the terminated and tailbiting cases of the three code examples for different spatial lengths  $T \in \{12, 21, 30, 39, 48, 57, 66, 75, 84, 300\}$ . We consider Gray-labeled PM-16-QAM, PM-64-QAM, and PM-256-QAM. In all scenarios, a WD is employed with a window size of  $W = 5$  and  $l_{\max} = 10$  iterations per window. The target BER for the optimization is set to  $10^{-5}$ . Setting a lower target BER (e.g.,  $10^{-15}$ ) has virtually no influence on the optimization outcome due to the steepness of the predicted DE performance curves (see Fig. 5). This assumes that there are no error floors due to harmful graph structures, which cannot be modeled using DE and are not considered in this paper. An analysis of the error floor for the considered SC-GLDPC codes is an interesting topic for future work and beyond the scope of this paper.

## 6.1 Structure of the Optimized Bit Mapper for Tailbiting Codes

For the tailbiting codes, the optimized bit mappers have an interesting structure, which is illustrated in Fig. 7 for two scenarios: (a) the SC-LDPC code with SDD, PM-64-QAM, and  $T = 30$ ; (b) the SC-GLDPC code with HDD,  $R' = 0.75$ , PM-16-QAM, and  $T = 30$ .

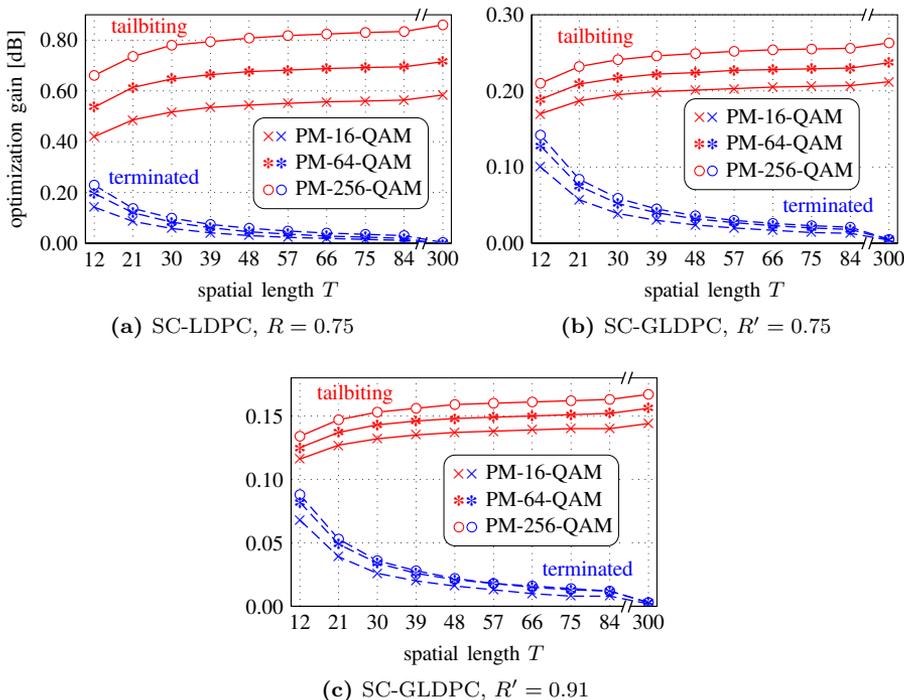
For PM-64-QAM and PM-16-QAM, the modulation bits have three and two different protection levels, respectively. Due to the tailbiting code structure, the bit allocation is invariant to a circular shift, assuming that the scheduling of the WD is modified according to the same shift. For the allocation shown in Fig. 7, it is assumed that the first decoding window begins at the first spatial position. The optimized bit mapper in both scenarios deviates significantly from the baseline mapper in the first few spatial positions. For the SC-LDPC code, the coded bits corresponding to the second, third, and fourth spatial position are proportionally more allocated to the best (green) and intermediate (yellow) protection level of PM-64-QAM. Similarly, for the SC-GLDPC code, the coded bits corresponding to the second and third spatial position are proportionally more allocated to the best modulation bit of PM-16-QAM. In both cases, the optimized allocation leads to a locally improved decoding convergence and initiates a wave-like decoding behavior comparable to that of terminated codes, i.e., the unequal error protection of the signal constellation is exploited to create an artificial termination boundary.

The performance gain due to the optimized bit mapper (which is quantified in the next section) comes at the expense of some increase in system complexity. In particular, one has to account for additional buffering because a symbol cannot be transmitted until all its  $m$  bits are encoded. For simplicity, let us assume a model where the FEC encoder outputs coded bits in blocks of  $MK'$  or  $m_c n_B/2$  bits, i.e., the number of bits per spatial position, and symbols are immediately modulated as soon as all  $m$  modulation bits are available. Then, no buffering is required for the sequential baseline mapper. On the other hand, the “worst-case” bit mapper allocates 100% of the coded bits in the first  $T/m$  spatial positions to the first modulation bit, 100% in the next  $T/m$  positions to the second bit, and so on (i.e.,  $a_{i,j} = 1$  for  $(i-1)T/m + 1 \leq j \leq iT/m$ ). Consequently, no bits are allocated to the last modulation bit until spatial position  $(m-1)T/m + 1$  and buffering of all coded bits up to position  $(m-1)T/m$  is required. In all considered scenarios, however, the required additional buffer size (in terms of the number of spatial positions) due to the optimized bit mappers did not exceed 2.

## 6.2 Optimization Gain

In Fig. 8, we show the optimization gain (in dB) as a function of the spatial length  $T$  for all considered scenarios. The optimization gain is defined as the difference between the decoding threshold using the baseline bit mapper and the decoding threshold using the optimized bit mapper. The gain quantifies the performance improvement one can expect by employing the optimized bit mappers assuming long codes.

Regardless of the signal constellation or code class, the optimization gain decreases with  $T$  for the terminated codes and increases for the tailbiting codes. This behavior can be explained as follows. The optimization gain for the tailbiting codes comes from allocating more coded bits in the beginning of the spatial chain to good modulation bits in order to initiate a decoding wave. This, however, reduces the effective capacity for the bits in the middle part of the spatial chain. As  $T$  increases, this reduction becomes



**Figure 8:** Optimization gain as a function of the spatial length  $T$ .

negligible and the optimization gain tends to a constant value. For terminated codes, a decoding wave is initiated by default and the optimized bit mapper increases the effective capacity for the bits in the middle part by allocating bits in the beginning and end of the chain proportionally more to modulation bits with lower protection levels. Again, as  $T$  increases, this effect becomes negligible and the gain approaches zero. As a result, while the tailbiting codes significantly benefit from the optimization, the gain for the considered terminated codes is limited, i.e., for  $T \geq 30$  the gain is  $< 0.1$  dB in all cases.

It can also be observed that the optimization gain generally depends on the signal constellation. The gain increases with the modulation order  $M$  due to the increased number of protection levels and stronger unequal error protection. This gain increase can also be observed when optimizing bit mappers for irregular LDPC codes, see, e.g., [36]. It is also important to stress that the optimization relies on the availability of a signal constellation with different protection levels in order to provide a performance gain. In particular, the techniques do not apply to PM-BPSK or (Gray-labeled) PM-QPSK.

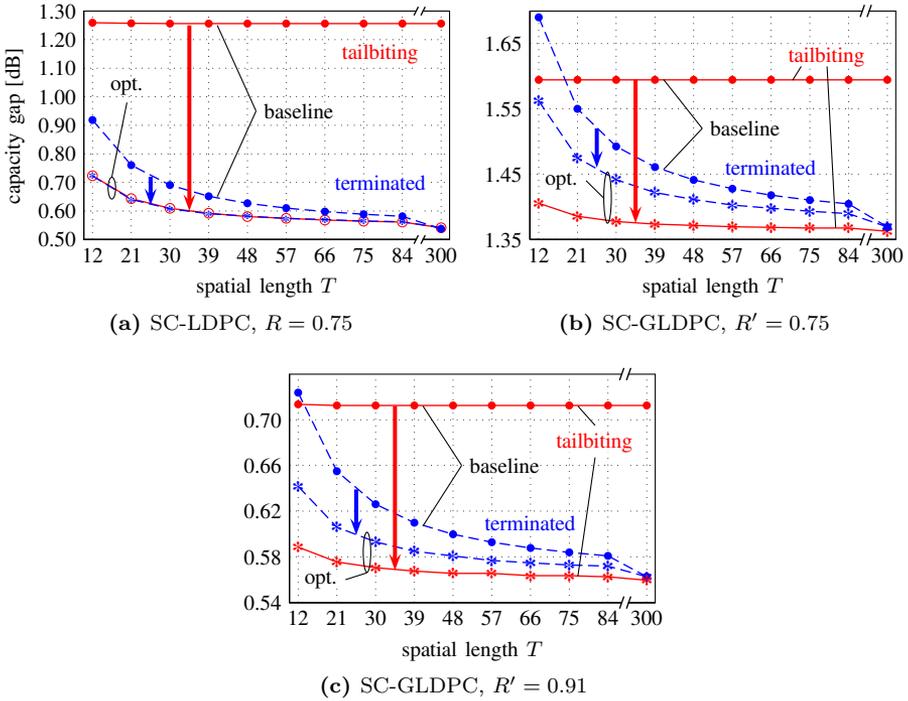
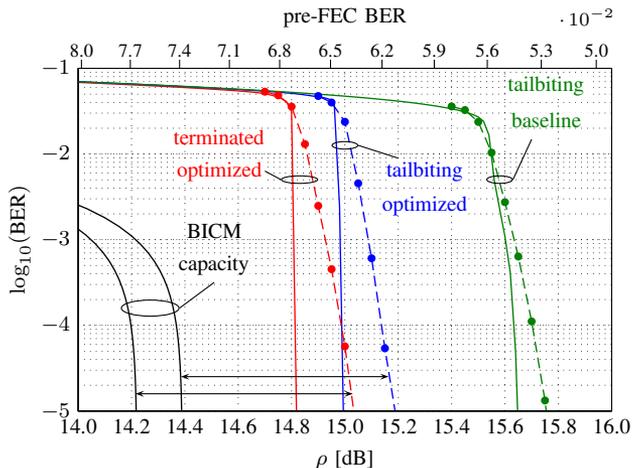


Figure 9: Capacity gap as a function of the spatial length  $T$  for PM-64-QAM.

### 6.3 Gap to Capacity

In order to gain some insight into the performance of the terminated and tailbiting codes relative to each other, the capacity gap (in dB) as a function of the spatial length  $T$  is shown in Fig. 9 for PM-64-QAM. For SDD of the SC-LDPC codes, the BICM capacity [5] is taken as a benchmark. For HDD of the GLDPC codes, the capacity of the BSC with averaged crossover probability is taken as a benchmark, similar to [37]. Alternatively, one may use the capacity of the sum of the  $m$  parallel BSCs as a benchmark, which is larger. The gains discussed in the previous subsection are indicated in Fig. 9 with arrows.

The decoding thresholds for the baseline systems are approximately independent of  $T$ . Therefore, the capacity gap for the tailbiting codes remains constant in all cases, while the capacity gap for the terminated codes decreases due to the vanishing rate loss. For the baseline systems, the performance difference between terminated and tailbiting codes is most significant for the SC-LDPC codes (up to 0.75 dB), while for the SC-GLDPC codes the difference is lower (up to 0.25 dB for  $R' = 0.75$  and up to 0.19 dB for  $R' = 0.91$ ). In all cases, the capacity gap is reduced by employing the optimized bit mappers. If we compare the optimized systems, it can be seen that the gap for the SC-LDPC codes is virtually identical for terminated and tailbiting cases. For the SC-GLDPC codes, the



**Figure 10:** Simulation results for the SC-LDPC code with PM-64-QAM over the AWGN channel. The codes have length 360 000.

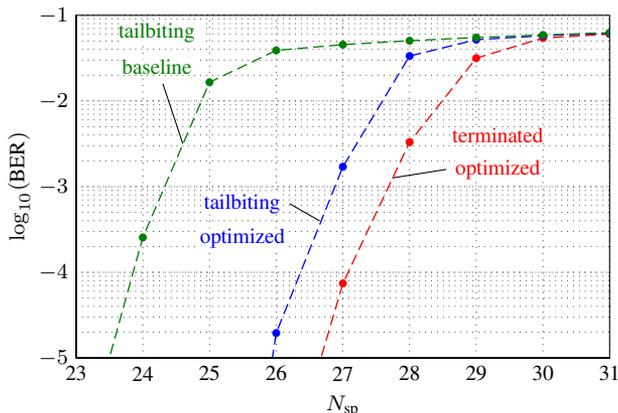
tailbiting codes perform closer to capacity, albeit the difference to the terminated codes for  $T \geq 30$  is small. For very long spatial lengths (i.e.,  $T = 300$ ), the capacity gap virtually overlaps also for the SC-GLDPC codes.

## 6.4 Simulation Results

The results presented in the previous two subsections are based on decoding thresholds, i.e., assume an infinite code length. The deviation of the DE analysis from the finite-length performance is determined by the lifting factor  $M$  and the number of CNs per position  $m_c$ , see Fig. 5.

As an example, consider the SC-LDPC code with  $T = 30$  and  $M = 3000$  leading to a code length of  $n_C = 360\,000$ . The rates are  $R(30) \approx 0.742$  and  $R = 0.75$ , respectively. In Fig. 10, we show simulation results (dashed lines with dots) and the analytical P-EXIT prediction (solid lines) for the AWGN channel, i.e., a linear transmission scenario, assuming PM-64-QAM. As predicted by the optimization gain in Fig. 8(a), the tailbiting code performs significantly better with an optimized bit mapper and a gain of  $\approx 0.55$  dB is achieved at a BER of  $10^{-5}$ . The terminated code performs better for the same SNR, but entails a smaller spectral efficiency due to the rate loss. The gap to the BER-constrained BICM capacity [22, p. 17] of the two optimized systems, as indicated by the arrows and predicted from Fig. 9(a), is approximately the same (as is the gap to the AWGN channel capacity, not shown).

Lastly, we also present simulation results for a nonlinear transmission scenario. We set  $\alpha = 0.25$  dB/km,  $\beta_2 = -21.668$  ps<sup>2</sup>/km,  $\gamma = 1.4$  W<sup>-1</sup> km<sup>-1</sup>,  $\nu_s = 1.934 \times 10^{14}$  Hz,  $n_{sp} = 1.622$ ,  $R_s = 40$  GBaud, and  $L_{sp} = 70$  km. A root-raised cosine pulse  $p(t)$



**Figure 11:** Simulation results for the SC-LDPC with PM-64-QAM over a dispersion uncompensated transmission link.

with a roll-off factor of 0.25 is used. We employ the symmetric split-step Fourier method with two samples per symbol and a fixed step size [38, Sec. 2.4.1]. The input power per polarization is set to  $P = -2.5$  dBm. In the simulation model, the polarization state is assumed to be known and perfect timing and carrier synchronization is assumed. In Fig. 11, the simulated BER of the PM transmission systems is plotted as a function of the number of fiber spans  $N_{sp}$ . For the tailbiting code, the 0.55 dB gain obtained by using the optimized bit mapper translates into an increase of the transmission reach by roughly 3 additional spans or approximately 13%. This gain is obtained at almost no increased system complexity cost, i.e., by simply replacing the baseline bit mapper with an optimized one. This reach extension can also be approximately calculated using the analytical expression for the SNR  $\rho$  as a function of the number of spans presented in [24]. The terminated code enables a longer transmission reach of approximately one span, at the expense of a 1.2% decrease in spectral efficiency. The performance of the terminated code with the baseline bit mapper is very close to the performance of the tailbiting code with the optimized bit mapper and is therefore not shown in Figs. 10 and 11 for clarity.

## 7 Conclusions

In this paper, we considered the optimized allocation of the coded bits from the FEC encoder to the modulation bits for terminated and tailbiting SC FEC schemes, assuming both SDD and HDD, as well as different signal constellations. Terminated SC codes generally benefit little from the optimization, particularly for long spatial lengths. However, the performance of tailbiting SC-LDPC codes can be significantly improved. With an optimized bit allocation, the terminated and tailbiting codes are competitive, in the sense that spectral efficiency can be traded for transmission reach, at approximately the

same gap to capacity.

## 8 Acknowledgements

The authors would like to acknowledge helpful discussions with Henry D. Pfister regarding the SC-GLDPC ensemble.

## References

- [1] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, “Capacity limits of optical fiber networks,” *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, Feb. 2010.
- [2] B. P. Smith and F. R. Kschischang, “Future prospects for FEC in fiber-optic communications,” *IEEE J. Sel. Topics. Quantum Electron.*, vol. 16, no. 5, pp. 1245–1257, Oct. 2010.
- [3] L. Schmalen, A. J. de Lind van Wijngaarden, and S. ten Brink, “Forward error correction in optical core and optical access networks,” *Bell Labs Tech. J.*, vol. 18, no. 3, pp. 39–66, Mar. 2013.
- [4] L. Beygi, E. Agrell, J. M. Kahn, and M. Karlsson, “Coded modulation for fiber-optic networks,” *IEEE Signal Processing Mag.*, vol. 31, no. 2, pp. 93–103, Mar. 2014.
- [5] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [6] A. Leven and L. Schmalen, “Status and recent advances on forward error correction technologies for lightwave systems,” in *Proc. European Conf. Optical Communication (ECOC)*, London, UK, 2013, p. We.2.C.1.
- [7] Y. Miyata, T. Sugihara, K. Kubo, H. Yoshida, and W. Matsumoto, “A spatially-coupled type LDPC code with an NCG of 12 dB for optical transmission beyond 100 Gb/s,” in *Proc. Optical Fiber Communication Conf. (OFC)*, Anaheim, USA, Mar. 2013, p. OM2B.4.
- [8] S. Kudekar, T. Richardson, and R. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [9] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

- [10] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [11] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progress Report 42-154, JPL*, 2005.
- [12] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "AWGN channel analysis of terminated LDPC convolutional codes," in *Proc. Information Theory and Applications Workshop (ITA)*, La Jolla, CA, 2011.
- [13] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [14] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN," *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.
- [15] A. Leven and L. Schmalen, "Status and recent advances on forward error correction technologies for lightwave systems," *J. Lightw. Technol.*, vol. 32, no. 16, pp. 2735–2750, Aug. 2014.
- [16] C. Häger, A. Graell i Amat, F. Brännström, A. Alvarado, and E. Agrell, "Improving soft FEC performance for higher-order modulations via optimized bit channel mappings," *Opt. Express*, vol. 22, no. 12, pp. 14 544–14 558, Jun. 2014.
- [17] M. Lentmaier and G. P. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Austin, TX, 2010, pp. 709–713.
- [18] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Matahreh, "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication," in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.
- [19] I. Djordjevic, O. Milenkovic, and B. Vasic, "Generalized low-density parity-check codes for optical communication systems," *J. Lightw. Technol.*, vol. 23, no. 5, pp. 1939–1946, May 2005.
- [20] I. Djordjevic, L. Xu, T. Wang, and M. Cvijetic, "GLDPC codes with Reed-Muller component codes suitable for optical communications," *IEEE Commun. Lett.*, vol. 12, no. 9, pp. 684–686, Sep. 2008.
- [21] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003.

- [22] W. Ryan and S. Lin, *Channel Codes Classical and Modern*. Cambridge University, 2009.
- [23] G. P. Agrawal, *Lightwave Technology: Telecommunication Systems*. Wiley-Interscience, 2005.
- [24] L. Beygi, E. Agrell, P. Johannisson, M. Karlsson, and H. Wymeersch, “A discrete-time model for uncompensated single-channel fiber-optical links,” *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3440–3450, Nov. 2012.
- [25] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, and F. Forghieri, “EGN model of non-linear fiber propagation,” *Opt. Express*, vol. 22, no. 13, pp. 16 335–16 362, Jun. 2014.
- [26] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, “Windowed decoding of protograph-based LDPC convolutional codes over erasure channels,” *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [27] M. B. Tavares, K. S. Zigangirov, and G. P. Fettweis, “Tail-biting LDPC convolutional codes,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Nice, Italy, 2007.
- [28] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [29] S. Ten Brink, “Convergence of iterative decoding,” *Electronics Letters*, vol. 35, no. 10, pp. 806–808, May 1999.
- [30] G. Liva and M. Chiani, “Protograph LDPC codes design based on EXIT analysis,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Washington, DC, 2007.
- [31] Y.-Y. Jian, “On the analysis of spatially-coupled GLDPC codes and the weighted min-sum algorithm,” Ph.D. dissertation, Texas A&M University, 2013.
- [32] B. P. Smith, “Error-correcting codes for fibre-optic communication systems,” Ph.D. dissertation, University of Toronto, 2011.
- [33] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Opt.*, vol. 11, pp. 341–359, Nov. 1997.
- [34] L. Schmalen and S. ten Brink, “Combining spatially coupled LDPC codes with modulation and detection,” in *Proc. Int. Conf. Systems, Communication and Coding (SCC)*, Munich, Germany, Jan. 2013.

- [35] L. M. Zhang and F. R. Kschischang, “Staircase codes with 6% to 33% overhead,” *J. Lightw. Technol.*, vol. 32, no. 10, pp. 1999–2002, May 2014.
- [36] T. Cheng, K. Peng, J. Song, and K. Yan, “EXIT-aided bit mapping design for LDPC coded modulation with APSK constellations,” *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 777–780, Jun. 2012.
- [37] B. Smith and F. R. Kschischang, “A pragmatic coded modulation scheme for high-spectral-efficiency fiber-optic communications,” *J. Lightw. Technol.*, vol. 30, no. 13, pp. 2047–2053, Jul. 2012.
- [38] G. P. Agrawal, *Nonlinear Fiber Optics*, 4th ed. Academic Press, 2006.

PAPER **C**

**On Parameter Optimization for Staircase Codes**

Christian Häger, Alexandre Graell i Amat, Henry D. Pfister,  
Alex Alvarado, Fredrik Brännström, and Erik Agrell

in *Proc. Optical Fiber Communication Conf. and Exposition (OFC)*,  
Los Angeles, CA, 2015

*The layout has been revised.*

## Abstract

We discuss the optimization of staircase code parameters based on density evolution. An extension of the original code construction is proposed, leading to codes with steeper waterfall performance.

## 1 Introduction

In [1], Smith *et al.* proposed a new class of forward error correction (FEC) codes for optical transport networks, named staircase codes, which offer a 0.42 dB net coding gain (NCG) improvement over the best code from the ITU-T G.975.1 recommendation, while at the same time maintaining a low implementation complexity [1]. Staircase codes are a class of generalized low-density parity-check (GLDPC) codes where all variable nodes in the underlying Tanner graph have degree two and the check nodes correspond to Bose–Chaudhuri–Hocquenghem (BCH) codes. The code construction described in [1, Sec. IV] is specified in terms of the BCH code parameters  $(\nu, t, s)$ , where  $\nu$  is the Galois field extension degree,  $t$  the error-correction capability, and  $s$  the shortening parameter. In [2], these parameters were optimized assuming FEC overheads (OHs) ranging from 6.25% to 33.33%. Given a predefined parameter space, the optimization is based on a brute-force search using extensive software simulations to predict the code performance. Furthermore, in order to reduce the optimization complexity, a simplified iterative hard-decision decoding (HDD) is assumed. This iterative HDD does not account for miscorrections in the bounded-distance decoding of the BCH codes.

In this paper, we discuss a different approach to finding good staircase code parameters for a fixed OH. We observe that, for a certain choice of parameters, staircase codes are contained in the spatially-coupled GLDPC (SC-GLDPC) code *ensemble* proposed in [3], whose asymptotic behavior can be analyzed using density evolution (DE). In particular, we use decoding thresholds (which are defined further below) as the optimization criterion. This approach offers a significantly reduced optimization time (or the possibility to explore a larger parameter space) and can rigorously account for miscorrections. The latter is obviously also possible with simulations, i.e., by performing bounded-distance decoding of the BCH codes, albeit at the cost of an increased optimization time.

A staircase code with parameters optimized with the above approach does not guarantee to provide the best performance for a given pre-FEC bit error rate (BER). This is because the performance of staircase codes is not necessarily well predicted by the asymptotic DE in [3], in particular for small staircase block sizes. To overcome this problem, we propose an extension of staircase codes with larger staircase block sizes by allowing for multiple code constraints per row/column in the staircase array. It is shown that the proposed codes have steeper waterfall performance curves compared to those obtained from the original construction and match the DE prediction more closely.

## 2 Staircase Codes

Let  $\mathcal{C}$  be a shortened primitive BCH code with (even) length  $n = 2^\nu - 1 - s$  and dimension  $k = 2^\nu - \nu t - 1 - s$ . A staircase code with rate  $R = 2k/n - 1$  (and  $\text{OH} = 1/R - 1$ ) is defined as the set of all matrix sequences  $\mathbf{B}_i \in \{0, 1\}^{a \times a}$ ,  $i = 0, 1, 2, \dots$ , such that the rows in  $[\mathbf{B}_{i-1}^T \mathbf{B}_i]$  for all  $i \geq 1$  form valid codewords of  $\mathcal{C}$ , where  $a = n/2$  and  $\mathbf{B}_0$  is initialized to the all-zero matrix. Similar to classical product codes, codewords in a staircase code are naturally represented as two-dimensional arrays and  $a$  is the size of one block in the characteristic staircase array shown in [1, Fig. 4].

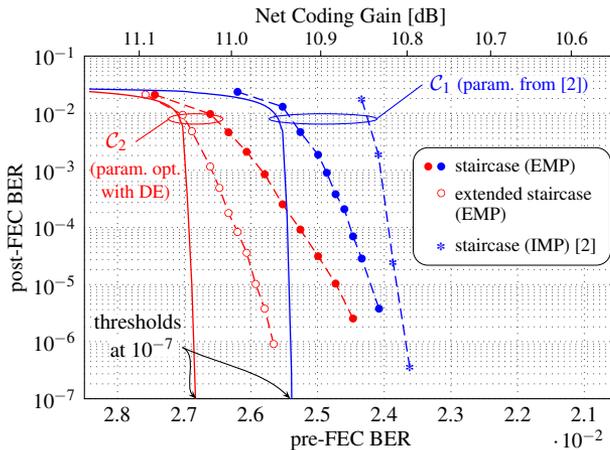
Motivated by the work on block-wise braided block codes [4], a SC-GLDPC code ensemble was proposed in [3]. This code ensemble is specified in terms of the parameters  $(\mathcal{C}, m, L, w)$ , where  $m$  is the number of constraint nodes per spatial position,  $L$  the total number of spatial positions, and  $w$  the coupling width. Staircase codes are closely related to block-wise braided block codes [5] and both code classes can be seen as particular codes in the ensemble described in [3, Def. 2] with a proper interleaver choice. In particular, it can be shown that the staircase code for a given  $\mathcal{C}$  is contained in the ensemble for  $m = n/2$ ,  $L \rightarrow \infty$ , and  $w = 2$ .

### 2.1 Iterative Hard-Decision Decoding

The decoding of staircase codes described in [1, Sec. IV-A] is performed in a sliding-window fashion by iterating between the BCH decoders for all rows and columns in the staircase array consisting of  $W$  received blocks for a maximum of  $l$  iterations. This decoding scheme can be seen as an iterative message-passing algorithm with “hard” (i.e., binary) messages in the Tanner graph describing the code. However, it is pointed out in [3] that the message-passing rule associated with the decoding algorithm in [1] violates the principle that only extrinsic messages should be exchanged during iterative decoding. Hence, the authors in [3] refer to this decoding as iterative HDD with *intrinsic* message passing (IMP). In this paper, we employ the iterative HDD with *extrinsic* message passing (EMP) proposed in [3], and adapted to operate in a sliding-window fashion for staircase codes. EMP can provide better performance compared to IMP [5], and its performance can be analyzed using DE, even in the event of miscorrection [3]. For more details about this algorithm and the differences with respect to the decoding in [1], we refer the reader to [3] and [5, Algorithm 1].

### 2.2 Density Evolution

DE is a tool to predict the iterative decoding performance of asymptotically long codes in the waterfall region. DE mimics the decoding process under a cycle-free graph assumption by tracking how the probability densities of the exchanged messages evolve with iterations. In [3], a DE analysis is presented for the  $(\mathcal{C}, m, L, w)$  SC-GLDPC code ensemble with iterative HDD and EMP, assuming that  $m \rightarrow \infty$ .



**Figure 1:** DE (solid) and simulation (dashed) results

*Example:* Let  $\mathcal{C}_1$  be a BCH code with parameters  $(\nu, t, s) = (9, 5, 151)$  [2, Table II]. For the  $(\mathcal{C}_1, \infty, 30, 2)$  SC-GLDPC code ensemble, if we adapt the DE analysis in [3, eq. (9)] to a sliding-window decoder with  $W = 7$  and  $l = 8$ , one obtains the solid blue curve in Fig. 1. The simulated performance (assuming a binary symmetric channel (BSC)) of the staircase code with  $\mathcal{C}_1$  (which leads to a block size of  $a = 180$  and  $\text{OH} = 33.33\%$ ) using EMP is shown in Fig. 1 by the dashed blue curve with dots. The performance using IMP is shown by the dashed blue curve with stars, where the data is taken from [2, Fig. 2].  $\triangle$

It can be observed that DE accurately predicts the pre-FEC BER region where the staircase code performance curve “bends” into the characteristic waterfall behavior. This motivates the use of DE to find good staircase code parameters. In particular, we use the decoding threshold (for a finite number of decoding iterations), which is defined as the pre-FEC BER value where the DE curve crosses a certain target post-FEC BER. Decoding thresholds can be computed numerically using a bisection search over a given pre-FEC BER range. The decoding threshold for the code ensemble in Example 1 at a post-FEC BER of  $10^{-7}$  is approximately given by  $2.54 \cdot 10^{-2}$  and also indicated in Fig. 1.

It can also be observed that the actual performance of the staircase code deviates quite significantly from the DE prediction once the pre-FEC BER is smaller than the decoding threshold. In this context, it is important to point out that the DE analysis applies to the SC-GLDPC ensemble for  $m \rightarrow \infty$ , whereas staircase codes belong to the ensemble where  $m$  is fixed to half the length of the BCH code. For a random code taken from the ensemble, the parameter  $m$  determines the steepness of the BER curve. Similar observations can be made also for staircase codes, where the steepness of the BER curves in the waterfall region is determined by the block size  $a$ , see [2, Fig. 2].

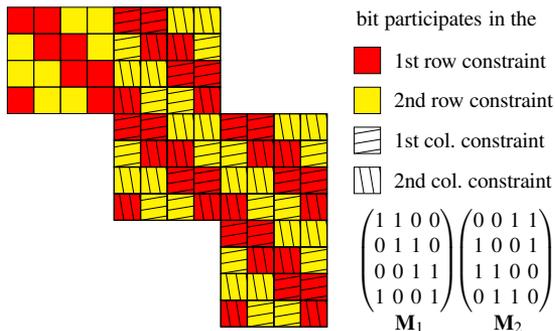


Figure 2: Modified staircase array

### 3 Extended Staircase Codes

By allowing for  $q > 1$  code constraints in each row and column of the staircase array, one can increase the block size of staircase codes to  $\tilde{a} = qa$ . In order to indicate which bit participates in the  $j$ -th code constraint of a particular row or column, one may use  $q$  masking matrices  $\mathbf{M}_j \in \{0, 1\}^{\tilde{a} \times \tilde{a}}$ ,  $j \in \{1, 2, \dots, q\}$ , with  $a$  ones in each row such that  $\sum_{j=1}^q \mathbf{M}_j$  is the all-one matrix. We informally write  $\mathbf{A} \cap \mathbf{M}_j$  to denote the  $\tilde{a} \times a$  matrix containing the elements from the  $\tilde{a} \times \tilde{a}$  matrix  $\mathbf{A}$  where the corresponding entry in  $\mathbf{M}_j$  is equal to one. The extended staircase code with larger block size is defined as the set of all matrix sequences  $\mathbf{B}_i \in \{0, 1\}^{\tilde{a} \times \tilde{a}}$ ,  $i = 0, 1, 2, \dots$ , such that the rows in  $[\mathbf{B}_{i-1}^T \cap \mathbf{M}_j \ \mathbf{B}_i \cap \mathbf{M}_j]$  for all  $i > 0$  and  $j$  form valid codewords in  $\mathcal{C}$ , where  $\mathbf{B}_0$  is the all-zero matrix. As an example, the staircase array for  $n = 4$  and  $q = 2$  is shown in Fig. 2. We remark that this extension is itself a special case of a more general technique, where the Tanner graph of the staircase code can be interpreted as a protograph which is lifted in order to obtain a larger graph [6]. Our particular choice for the lifting factor (i.e.,  $q^2$ ) and type of lifting preserves many properties of the original staircase codes, e.g., the staircase array structure and time-invariant encoding/decoding operations.

### 4 Results and Discussion

We consider the same parameter space as in [2], i.e., the product set of  $\text{OH} \in \{1/i : i = 3, 4, \dots, 16\}$ ,  $\nu \in \{8, 9, 10, 11, 12\}$ , and  $t \in \{2, 3, 4, 5, 6\}$ . In Table 1, we list the parameters that give the best decoding threshold at a BER of  $10^{-15}$  for the SC-GLDPC ensemble with  $L = 30$  and  $w = 2$  according to the DE analysis accounting for miscorrection [3, eq. (9)] and adapted to a sliding-window decoder with  $W = 7$  and  $l = 8$ . We also show the NCG at a BER of  $10^{-15}$  using the obtained decoding thresholds and the corresponding gap to the maximum NCG for the BSC. To allow for a direct comparison, we repeat the values that are reported in [2], where results for  $\text{OH} = 8.33\%$  are unfortunately

OH [%]	6.25	6.67	7.14	7.69	8.33	9.09	10.00	11.11	12.50	14.29	16.67	20.00	25.00	33.33
$\nu$	10 11	10 11	10 11	10 11	10 N/A	10 10	10 10	10 10	9 10	9 10	9 9	9 9	8 9	8 9
$t$	3 4	3 5	3 5	3 5	3 N/A	4 4	4 4	5 4	3 4	3 4	4 4	4 4	3 5	3 5
$s$	3 551	63 287	123 397	183 507	243 N/A	63 63	143 143	23 223	25 303	79 383	7 7	79 79	15 61	63 151
$a$	510 748	480 880	450 825	420 770	390 N/A	480 480	440 440	500 400	243 360	216 320	252 252	216 216	120 225	96 180
NCG [dB]	9.50 9.44	9.56 9.54	9.62 9.60	9.69 9.66	9.76 N/A	9.89 9.82	9.97 9.87	10.08 9.96	10.21 10.05	10.32 10.13	10.52 10.32	10.64 10.41	10.88 10.62	11.03 10.70
Gap [dB]	0.39 0.45	0.41 0.43	0.43 0.45	0.45 0.48	0.48 N/A	0.45 0.52	0.48 0.59	0.51 0.63	0.53 0.68	0.58 0.77	0.58 0.78	0.69 0.92	0.74 1.00	0.95 1.28

Table 1: Optimized staircase code parameters based on thresholds (top) and taken from [2] (bottom)

not available. We would like to stress that in [2], software simulations are extrapolated down to  $10^{-15}$  to estimate the NCG and gap. Hence, those values are not directly comparable to the ones we obtain by using decoding thresholds. Rather, the threshold results should be seen as a way to identify further room for improvement, in particular using the extended staircase code construction. As an example, for OH = 33.33%, we simulated the staircase code based on the BCH code  $\mathcal{C}_2$  with  $(\nu, t, s) = (8, 3, 63)$  together with the extended staircase code for  $q = 2$ . The results are shown in Fig. 1 by the dashed red lines with dots and circles, respectively. The extended staircase code has a steeper waterfall performance at the expense of a larger block size of  $\tilde{a} = 192$ . The block size is, however, comparable to that of the staircase code with  $\mathcal{C}_1$  and, extrapolating the simulation results to  $10^{-15}$ , we expect a gain of approximately 0.11 dB. If larger block sizes can be tolerated, further improvements (up to around 0.33 dB) can be expected by increasing  $q$ .

## 5 Conclusions

We have shown that the DE analysis in [3] can be used as an effective tool for finding good staircase code parameters. Compared to the simulation-based method in [2], this approach is less complex (e.g., the parameter space in [2] can be searched within seconds) and can account for miscorrections assuming iterative HDD with EMP. An extended staircase code construction was proposed with steeper waterfall performance at the expense of a larger staircase block size.

## References

- [1] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase codes: FEC for 100 Gb/s OTN,” *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.
- [2] L. M. Zhang and F. R. Kschischang, “Staircase codes with 6% to 33% overhead,” *J. Lightw. Technol.*, vol. 32, no. 10, pp. 1999–2002, May 2014.
- [3] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high rates with iterative hard-decision decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [4] A. J. Feltström, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, “Braided block codes,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jul. 2009.
- [5] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Mazahreh, “Iterative hard-decision decoding of braided BCH codes for high-speed optical communication,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.

- [6] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN Progress Report 42-154*, JPL, 2005.



PAPER **D** 

**Density Evolution for Deterministic Generalized Product Codes on the  
Binary Erasure Channel**

Christian Häger, Henry D. Pfister, Alexandre Graell i Amat, and Fredrik Brännström,

submitted to *IEEE Trans. Inf. Theory*

*The layout has been revised.*

## Abstract

Generalized product codes (GPCs) are extensions of product codes (PCs) where code symbols are protected by two component codes but not necessarily arranged in a rectangular array. We consider a deterministic construction of GPCs (as opposed to randomized code ensembles) and analyze the asymptotic performance over the binary erasure channel under iterative decoding. Our code construction encompasses several classes of GPCs previously proposed in the literature, such as irregular PCs, block-wise braided codes, and staircase codes. It is assumed that the component codes can correct a fixed number of erasures and that the length of each component code tends to infinity. We show that this setup is equivalent to studying the behavior of a peeling algorithm applied to a sparse inhomogeneous random graph. Using a convergence result for these graphs, we derive the density evolution equations that characterize the asymptotic decoding performance. As an application, we discuss the design of irregular GPCs employing a mixture of component codes with different erasure-correcting capabilities.

## 1 Introduction

Many code constructions are based on the idea of building longer codes from shorter ones [1–3]. In particular, product codes (PCs), originally introduced by Elias in 1954 [4], are constructed from two linear component codes,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , with respective lengths  $n_1$  and  $n_2$ . The codewords in a PC are rectangular  $n_1 \times n_2$  arrays such that every row is a codeword in  $\mathcal{C}_1$  and every column is a codeword in  $\mathcal{C}_2$ . In 1981, Tanner significantly extended this construction and introduced generalized low-density parity-check (GLDPC) codes [5]. GLDPC codes are defined via bipartite graphs where variable nodes (VNs) and constraint nodes (CNs) represent code symbols and component code constraints, respectively. If the underlying graph of a GLDPC code consists exclusively of degree-2 VNs (i.e., each code symbol is protected by two component codes), the code is referred to as a generalized PC (GPC). Most of the examples presented in [5] fall into this category.

PCs have an intuitive iterative decoding algorithm and are used in a variety of applications [6,7]. In practice, the component codes are typically Bose–Chaudhuri–Hocquenghem (BCH) or Reed–Solomon codes, which can be efficiently decoded via algebraic bounded-distance decoding (BDD). This makes GPCs particularly suited for high-speed applications due to their significantly reduced decoding complexity compared to message-passing decoding of low-density parity-check (LDPC) codes [8]. For example, GPCs have been investigated by many authors as practical solutions for forward-error correction in fiber-optical communication systems [8–15].

The iterative decoding of GPCs is a standard element in many of these systems and the analysis of iterative decoding is typically based on density evolution (DE) [16, 17] using an ensemble argument. That is, rather than analyzing a particular code directly, one considers a set of codes, defined via suitable randomized connections between VNs and CNs in the Tanner graph. Some notable exceptions include Gallager’s original analysis based on deterministic constructions of large-girth LDPC codes [18], Tanner’s analysis of Hamming GPCs [5], the analysis of PCs using monotone graph properties [19], and the analysis of PCs based on the  $k$ -core problem [9, 11].

In this paper, we focus on the asymptotic performance of GPCs over the binary erasure channel (BEC) assuming iterative decoding based on BDD of the component codes. In particular, we consider the case where the component codes have a fixed erasure-correcting capability and the length of each component code tends to infinity. Like [9, 11, 19], we consider a *deterministic* construction of GPCs. Indeed, many classes of GPCs have a very regular structure in terms of their Tanner graph and are not at all random-like. The code construction we consider is sufficiently general to recover several of these classes as special cases, such as irregular PCs [20, 21], block-wise braided codes [22, Sec. III], and staircase codes [8]. The main contribution of this paper is to show that, analogous to DE for code ensembles, the asymptotic performance of the considered GPC construction is rigorously characterized by a recursive update equation.

Like [9, 11, 19], this paper is largely based on results that have been derived in random graph theory. In our case, the Tanner graph itself is deterministic and consists of a fixed arrangement of (degree-2) VNs and CNs. Randomness is introduced entirely due to the channel by forming the so-called residual graph (or error graph) from the Tanner graph, i.e., after removing known VNs and collapsing erased VNs into edges [9, 11, 19]. Thus, different channel realizations give rise to an ensemble of residual graphs, facilitating the analysis. The code construction considered here is such that the residual graph ensemble corresponds to the sparse inhomogeneous random graph model in [23]. Analyzing the decoding failure of the iterative decoder (for a fixed number of iterations) can then be translated into a graph-theoretic question about the behavior of a peeling algorithm applied to such a random graph. We can then use a convergence result in [23] to conclude that, as the number of vertices in the graph tends to infinity, the correct limiting behavior is obtained by evaluating the peeling algorithm on a multi-type branching process.

A similar connection between large random graphs and branching processes also arises in the DE analysis for code ensembles, e.g., irregular LDPC codes. The main difference between this and our setup is that, for code ensembles, the Tanner graph itself is random due to the randomized edge connections in the ensemble definition. DE relies on the fact that the asymptotic behavior of an extrinsic iterative message-passing decoder can be analyzed by considering an ensemble of computation trees [24, Sec. 3.7.2] (see also [25, Sec. 1]). This tree ensemble can alternatively be viewed as a multi-type branching process, where types correspond to VNs and CNs of different degrees. A tree-convergence and concentration result ensures that the performance of a code taken (uniformly at

random) from the ensemble will be close to the predicted DE behavior, provided that the code is sufficiently long [17, Th. 2].

The above ensemble approach can be applied to GLDPC codes and thus also to GPCs. For example, in [26], [27], [28] a DE analysis for protograph-based braided codes is presented, where the Tanner graph of a tightly-braided code is interpreted as a protograph [29]. An ensemble approach has been further applied to regular GPCs in [30], where the authors analyze the asymptotic ensemble performance and derive the corresponding iterative decoding thresholds. In [31–33], the authors perform a DE analysis for GPC ensembles paying special attention to so-called spatially-coupled codes. However, given the fairly regular Tanner graph structure of several GPC classes, it would be highly desirable to make precise statements about the performance of actual codes, without resorting to an ensemble argument.

The work here is closely related to [9, 11, 19]. In [19], combinatorial tools from the study of random graphs are used to analyze the iterative decoding of PCs. In [9], the authors point out the direct connection between the iterative decoding of PCs and a well-studied problem in random graph theory: the emergence of a  $k$ -core, defined as the largest induced subgraph where all vertices have degree at least  $k$  [34]. Indeed, assuming that all component codes can correct  $t$  erasures and allowing for an unrestricted number of iterations, the decoding either finishes successfully, or gets stuck and the resulting graph corresponds to the  $(t + 1)$ -core of the residual graph. The results in [34] apply to PCs only after some modifications (described in [9]), since the random graph model in [34] is slightly different than the actual one corresponding to the residual graph ensemble of PCs. In a later paper, Justesen considered GPCs for which the Tanner graph is based on a complete graph [11] (see, e.g., Fig. 1(b)). In that case, the results in [34] are directly applicable. The resulting codes are referred to as half-product codes (HPCs). Even though these codes have received very little attention in the literature, Tanner already used a similar construction [5, Fig. 6].

We use HPCs as the starting point for our analysis. The reason is that the residual graph of an HPC corresponds exactly to an instance of the Erdős–Rényi random graph model  $\mathcal{G}(n, p)$  [35, 36], which is arguably one of the most well-studied random graph models and also considerably simpler than the inhomogeneous random graph model in [23]. It is therefore instructive to consider this case in sufficient detail before analyzing generalizations to other GPCs. Even though other classes of GPCs are mentioned and discussed also in [11] (e.g., braided codes), so far, rigorous analytical results about the asymptotic performance of deterministic GPCs have been limited to conventional PCs and HPCs.

As an application of the derived DE equations for deterministic GPCs, we discuss the optimization of component code mixtures for HPCs. In particular, we consider the case where the component codes can have different erasure-correcting capabilities. It is shown that, similar to irregular PCs [20, 21], HPCs greatly benefit from employing component codes with different strengths. We further derive upper and lower bounds on the iterative

decoding thresholds of HPCs with component code mixtures. The upper bound is shown to have a graphical interpretation in terms of areas related to the DE equations, similar to the area theorem of irregular LDPC codes.

The remainder of the paper is structured as follows. We start by analyzing HPCs in Sections 2, 3, and 4. In particular, in Section 2 we discuss the code construction, the decoding algorithm, and state the main result about the asymptotic performance of HPCs in Theorem 1. In Section 3, we review the necessary background about random graphs and branching processes related to the proof of Theorem 1, which is then given in Section 4. In Section 5, we extend Theorem 1 to a general deterministic construction of GPCs and derive the corresponding DE equations. The optimization of component code mixtures for irregular HPCs is studied in Section 7. The paper is concluded in Section 8.

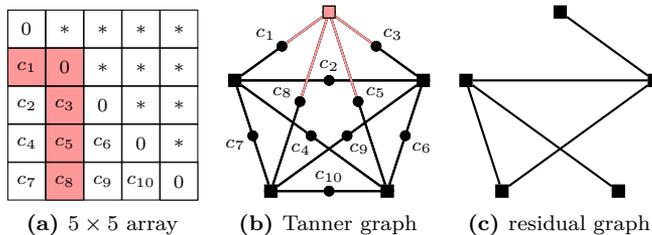
## 1.1 Notation

The following notation is used throughout the paper. We define the sets  $[n] \triangleq \{1, 2, \dots, n\}$ ,  $\mathbb{N}_0 \triangleq \{0, 1, 2, \dots\}$ , and  $\mathbb{N} \triangleq \{1, 2, \dots\}$ . The cardinality of a set  $\mathcal{A}$  is denoted by  $|\mathcal{A}|$ . Sequences are denoted by  $(x_n)_{n \geq 1} = x_1, x_2, \dots$ . The probability density function (PDF) of a random variable (RV)  $X$  is denoted by  $f_X(\cdot)$ . Expectation and probability are denoted by  $\mathbb{E}[\cdot]$  and  $\mathbb{P}(\cdot)$ , respectively. We write  $X \sim \mathbf{B}(p)$  if  $X$  is a Bernoulli RV with success probability  $p$ ,  $X \sim \mathbf{Bin}(n, p)$  if  $X$  is a Binomial RV with parameters  $n$  and  $p$ , and  $X \sim \mathbf{Po}(\lambda)$  if  $X$  is a Poisson RV with mean  $\lambda$ . With some abuse of notation, we write, e.g.,  $\mathbb{P}(\mathbf{Po}(\lambda) \geq t)$  for  $\mathbb{P}(X \geq t)$  with  $X \sim \mathbf{Po}(\lambda)$ . We define the Poisson tail probability as  $\Psi_{\geq t}(\lambda) \triangleq \mathbb{P}(\mathbf{Po}(\lambda) \geq t) = 1 - \sum_{i=0}^{t-1} \Psi_{=i}(\lambda)$ , where  $\Psi_{=i}(\lambda) \triangleq \frac{\lambda^i}{i!} e^{-\lambda}$ . We use boldface to denote vectors and matrices (e.g.,  $\mathbf{a}$  and  $\mathbf{A}$ ). Matrix transpose is denoted by  $(\cdot)^\top$ . Convergence in distribution (weak convergence) is denoted by  $\xrightarrow{d}$  and convergence in probability by  $\xrightarrow{\mathbb{P}}$ . For positive real functions, standard asymptotic notation (as  $n \rightarrow \infty$ ) will be used, e.g., we write  $f(n) = \mathcal{O}(g(n))$  if there exist constants  $k, n_0$  such that  $f(n) \leq kg(n)$  for all  $n > n_0$ . We write  $f(n) = \Omega(g(n))$  if there exist constants  $k, n_0$  such that  $f(n) \geq kg(n)$  for all  $n > n_0$ . We write  $f(n) = \Theta(g(n))$  if both  $f(n) = \mathcal{O}(g(n))$  and  $f(n) = \Omega(g(n))$ . Finally, a code is called an  $(n, k, d)$  code if it is linear and it has length  $n$ , dimension  $k$ , and minimum distance  $d$ .

## 2 Half-Product Codes

### 2.1 Code Construction

Let  $\mathcal{C}$  be a binary  $(n, k_{\mathcal{C}}, t+1)$  code and recall that such a code can correct all erasure patterns up to weight  $t$ . An HPC is constructed as follows (cf. [11, Sec. III-B]). Start with a conventional PC defined as the set of  $n \times n$  arrays such that each row and column is a codeword in the component code  $\mathcal{C}$ . Then, form a subcode of this PC by retaining only symmetric codeword arrays (i.e., arrays that are equal to their transpose) with a



**Figure 1:** Illustrations for an HPC with  $n = 5$ . In the array, “\*” means “equal to the transposed element”. The highlighted array elements illustrate one particular code constraint, which is also highlighted in the Tanner graph.

zero diagonal. After puncturing the diagonal and the upper (or lower) triangular part of the array, one obtains an HPC of length  $m = \binom{n}{2}$ . The Tanner graph representing an HPC is obtained from a complete graph with  $n$  vertices by interpreting each vertex as a CN corresponding to  $\mathcal{C}$  (shortened by one bit) and replacing each of the  $m$  edges by two half-edges joint together by a VN [11, Sec. III-B].<sup>1</sup> In the following, we assume some fixed (and arbitrary) ordering on the CNs and VNs.

*Example 1.* Figs. 1(a) and (b) show the code array and Tanner graph of an HPC for  $n = 5$  and  $m = 10$ . The highlighted array elements show the code symbols participating in the second row constraint, which, due to the enforced symmetry, is also the second column constraint. Effectively, each component code acts on an L-shape in the array, i.e., both a partial row and column, which includes the array diagonal. The degree of each CN is  $n - 1 = 4$ , due to the zeros on the diagonal. For example, for the highlighted CN in Fig. 1(b), the second bit position of  $\mathcal{C}$  is shortened (i.e., set to zero). Different bit positions are shortened for different CNs. Thus, the effective  $(n - 1, k_{\mathcal{C}} - 1, t + 1)$  component codes associated with the CNs are not necessarily the same.  $\triangle$

*Remark 1.* Recall that for a Tanner graph with generalized CNs, the edges emanating from each CN should also be labeled with the corresponding component code bit positions [5, Sec. II]. For HPCs, this assignment is implicitly given due to the array description. For example, the edges emanating from the highlighted CN in Fig. 1(b) correspond to bit positions 1, 5, 4, and 3 (in left-to-right order). Reshuffling these assignments may result in an overall code with different properties (e.g., rate) even though the Tanner graph remains unchanged [5, Sec. II], [11, Sec. III-A]. However, for the considered iterative decoder, the performance remains identical as long as the component code associated with each CN is able to correct  $t$  erasures, regardless of the bit position assignment.

We consider the limit  $n \rightarrow \infty$ , i.e., we use the number of CNs in the Tanner graph to

<sup>1</sup>One way to see this is to incorporate the symmetry constraint into the Tanner graph of a PC by connecting each VN to the “transposed” VN through a single parity-check (forcing the two to be equal). The graph now consists of degree-3 VNs (one row, one column, and one symmetry constraint), but can be simplified by removing all row (or column) constraints.

denote the problem size as opposed to the code length  $m = \mathcal{O}(n^2)$ . Assuming that  $\mathcal{C}$  has a fixed erasure-correcting capability<sup>2</sup>, this limit is sometimes referred to as the high-rate scaling limit or high-rate regime [31]. Indeed, if  $\mathcal{C}$  has dimension  $k_{\mathcal{C}}$ , the rate of an HPC is lower-bounded by [7, Sec. 5.2.1] (see also [5, Th. 1])

$$R \geq 1 - \frac{n(n-1 - (k_{\mathcal{C}} - 1))}{m} = 1 - 2\frac{n - k_{\mathcal{C}}}{n-1}. \quad (\text{D.1})$$

For a fixed erasure-correcting capability, we can assume that  $n - k_{\mathcal{C}}$  in (D.1) stays constant. It follows that  $R \rightarrow 1$  as  $n \rightarrow \infty$ . Note that the dimension of an HPC is  $k_{\mathcal{C}}(k_{\mathcal{C}} - 1)/2$  [11, Sec. III-B], [37, Lem. 8], which leads to a slightly larger rate than the lower bound in (D.1).

## 2.2 Binary Erasure Channel

Suppose that a codeword of an HPC is transmitted over the BEC with erasure probability  $p$ . Let  $I_k$  be the number of initial erasures associated with the  $k$ -th component code constraint. Due to symmetry, we have  $\mathbb{E}[I_k] = p(n-1)$  for all  $k \in [n]$ . Moreover, using a Chernoff bound, it can be shown that  $I_k$  concentrates around its mean (see, e.g., [19, Sec. IV]). As a consequence, for a fixed  $p > 0$  and  $n \rightarrow \infty$ , we see that any decoding attempt will be futile since  $\mathbb{E}[I_k] \rightarrow \infty$  for all  $k$ , but, on the other hand, we assumed a finite erasure-correcting capability for the component codes. We therefore let the erasure probability decay slowly as  $p = c/n$ , for a fixed  $c > 0$ . Since now  $p \rightarrow 0$  as  $n \rightarrow \infty$ , one may (falsely) conclude that decoding will always be successful in the asymptotic limit. As we will see, however, the answer depends crucially on the choice of  $c$ . It is thus instructive to interpret  $c$  as the “effective” channel quality for the chosen scaling of the erasure probability. From the above discussion, its operational meaning is given in terms of the expected number of initial erasures per component code constraint for large  $n$ , i.e.,  $\mathbb{E}[I_k] = c(n-1)/n \approx c$ .

*Remark 2.* One may alternatively assume a fixed erasure probability  $p$ , in conjunction with sequences of component codes that can correct a fixed fraction of erasures in terms of their block length. However, in that case, a simple analysis reveals that the (half-)product construction is essentially useless in the limit  $n \rightarrow \infty$ , and it is indeed better to just use the component code by itself (see the discussion in [19, Sec. IV]).

## 2.3 Iterative Decoding

Suppose decoding is performed iteratively for  $\ell$  iterations according to the following procedure. In each iteration, perform BDD for all CNs based on the values of the connected VNs. Afterwards, update previously erased VNs according to the decoding outcome. Updates are performed whenever there exists at least one CN where the weight

<sup>2</sup>More precisely, we consider sequences of codes with increasing length and fixed erasure-correcting capability.

of the associated erasure pattern is less than or equal to  $t$ . If the weight exceeds  $t$ , we say that the corresponding component code declares a decoding failure.

*Remark 3.* The decoding can alternatively be interpreted as an (intrinsic) message-passing decoder. In the first iteration, all VNs forward the received channel observations to the connected CNs. Then, CNs perform BDD based on all incoming messages and update their outgoing messages according to the decoding outcome. In subsequent iterations, outgoing VN messages are changed from erased to known if any of the two incoming CN messages becomes known. These update rules for VN and CN messages are not extrinsic (cf. [24, p. 117]), since the outgoing message along an edge may depend on the incoming message along the same edge.

An efficient way to represent the decoding is to consider the following peeling procedure. First, form the residual graph from the Tanner graph by deleting VNs and adjacent edges associated with correctly received bits and collapsing erased VNs into edges [9, 11, 19]. Then, in each iteration, determine all vertices that have degree at most  $t$  and remove them, together with all adjacent edges. The decoding is successful if the resulting graph is empty after (at most)  $\ell$  iterations.

*Example 2.* Fig. 1(c) shows the residual graph for the HPC in Example 1, where  $c_2$ ,  $c_3$ ,  $c_4$ ,  $c_7$ , and  $c_9$  are assumed to be erased. One may check that for  $t = 1$ , the decoding gets stuck after one iteration while for  $t = 2$ , the decoding finishes successfully after two iterations.  $\triangle$

*Remark 4.* The above parallel peeling procedure should not be confused with the sequential “peeling decoder” described in, e.g., [24, p. 117]. That decoder uses a different scheduling where vertices are removed sequentially and not in parallel, i.e., in each step one picks only one vertex with degree at most  $t$  (uniformly at random) and removes it [24, p. 117].

## 2.4 Asymptotic Performance

For a fixed  $\ell$ , we wish to characterize the asymptotic decoding performance as  $n \rightarrow \infty$ . We start by giving a heuristic argument behind the result stated in Theorem 1 below. For a similar discussion in the context of cores in random graphs, see [34, Sec. 2].

- Consider a randomly chosen CN. The decoding outcome of the BDD for this CN after  $\ell$  iterations depends only on the depth- $\ell$  neighborhood<sup>3</sup> of the vertex in the residual graph corresponding to this CN. The residual graph itself is an instance of the Erdős–Rényi random graph model  $\mathcal{G}(n, p)$ , which consists of  $n$  vertices. An edge between two vertices exists with probability  $p = c/n$ , independently of all other edges.

<sup>3</sup>The depth- $\ell$  neighborhood of a vertex is the subgraph induced by all vertices that can be reached by taking  $\ell$  or fewer steps from the vertex.

- For large  $n$ , the fixed-depth neighborhood approximately looks like a Poisson branching process, which starts with an initial vertex at depth 0 that has a Poisson number of neighboring vertices with mean  $c$  that extend to depth 1. Each of these vertices has again a Poisson number of neighboring vertices, independently of all other vertices, and so on.
- For large  $n$  and fixed  $\ell$ , one would therefore expect the probability that an individual CN declares a failure to be close to the probability that the root vertex of the first  $\ell$  generations of the branching process survives the same peeling procedure as described for the residual graph. We define the latter probability as  $z^{(\ell)}$ . We will see in Section 4.3 that

$$z^{(\ell)} = \Psi_{\geq t+1}(cx^{(\ell-1)}), \quad (\text{D.2})$$

where the function  $\Psi_{\geq t}$  is defined in Section 1.1 and  $x^{(\ell)}$  is defined recursively by  $x^{(0)} = 1$  and

$$x^{(\ell)} = \Psi_{\geq t}(cx^{(\ell-1)}). \quad (\text{D.3})$$

The main result for HPCs is as follows.

**Theorem 1.** *Let  $W_k$  be the indicator RV for the event that the  $k$ -th component code declares a decoding failure after  $\ell$  iterations of decoding and let the fraction of failed component codes be  $W = \frac{1}{n} \sum_{k=1}^n W_k$ . Then, we have*

$$\lim_{n \rightarrow \infty} \mathbb{E}[W] = z^{(\ell)}. \quad (\text{D.4})$$

Furthermore, for any  $\varepsilon \geq 0$ , there exist  $\delta > 0$ ,  $\beta > 0$ , and  $n_0 \in \mathbb{N}$  such that for all  $n > n_0$  we have

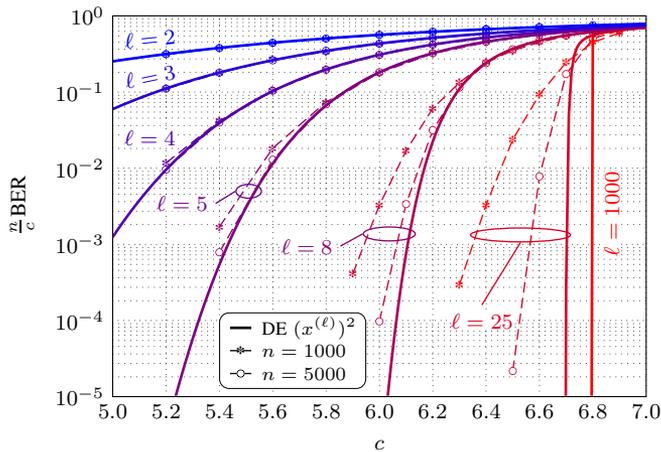
$$\mathbb{P}(|W - \mathbb{E}[W]| \geq \varepsilon) \leq e^{-\beta n^\delta}. \quad (\text{D.5})$$

*Proof.* The proof is given in Section 4. □

*Remark 5.* In our notation, we largely suppress the dependence of the involved RVs on  $n$  and  $\ell$  (e.g., one could write  $W^{(n, \ell)}$  instead of  $W$ ).

Combining (D.4) and (D.5) allows us to conclude that the code performance after  $\ell$  iterations (measured in terms of the RV  $W$ , i.e., the fraction of component codes that declare failure) converges almost surely to a deterministic value, i.e., it sharply concentrates around  $z^{(\ell)}$  for sufficiently large  $n$ . This result is analogous to the DE analysis of LDPC codes [17, Th. 2], and hence, we refer to (D.2) and (D.3) as the DE equations.

The chosen performance measure in Theorem 1 is the most natural one for the proof in Section 4. It is, however, possible to relate (D.2) and (D.3) to other performance measures that are more relevant in practice.



**Figure 2:** DE and simulation results for HPCs with  $t = 4$  as a function of the iteration number  $\ell$ .

*Example 3.* The meaning of the quantity  $x^{(\ell)}$  is given in Section 4.3 in terms of the Poisson branching process. The operational meaning in the coding context is as follows. Consider a randomly chosen erased bit. Asymptotically,  $x^{(\ell)}$  corresponds to the probability that the bit is not recovered after  $\ell$  decoding iterations by one of the two corresponding component codes. Since each bit is protected by two component codes, the overall probability of not recovering the bit is asymptotically given by  $(x^{(\ell)})^2$ . In Fig. 2, we plot the resulting DE prediction  $(x^{(\ell)})^2$  as a function of  $c$  for  $t = 4$  and different values of  $\ell$ , together with simulation results of the (scaled) bit error rate (BER) for  $n = 1000$  and  $n = 5000$ . Asymptotically as  $n \rightarrow \infty$ , we expect the simulation results to converge to the solid lines.  $\triangle$

Theorem 1 can be seen as an application of [23, Th. 11.6], except for the concentration bound in (D.5). In fact, [23, Th. 11.6] applies to a more general class of inhomogeneous random graphs, and we use it later when studying generalizations to other GPCs. The reason for including a separate proof for HPCs in Section 4 is two-fold. First, since [23, Th. 11.6] applies to a more general class of random graphs, it is instructive to consider the simplest case, i.e., the random graph  $\mathcal{G}(n, p)$  corresponding to HPCs, separately and in more detail. Second, rather than relying on [23, Th. 11.6], a self-contained proof of Theorem 1 allows us to point out similarities and differences to the DE analysis for LDPC codes in [17, 25], which we believe many readers are familiar with.

As mentioned in [11], iterative decoding of HPCs over the BEC is closely related to the emergence of a  $k$ -core in  $\mathcal{G}(n, p)$ . First observe that the overall decoding is successful if the RV  $W$  is strictly zero, i.e., if none of the component decoders declare failure. The existence of a core can then be related to the overall decoding failure assuming

an unrestricted number of iterations. Therefore, there is a subtle difference between studying the core and the overall decoding failure in our setup. In our case, the notion of decoding failure is always linked to the number of decoding iterations, which is assumed to be fixed (cf. [24, Sec. 3.19]). As a consequence, even though the overall decoding may fail after a finite number of iterations, there need not be a core in the residual graph. (The decoding may have been successful if we had done one more iteration, say.) Linking the decoding failure to the number of iterations has the advantage that it can always be determined locally (within the neighborhood of each vertex), whereas the core is a global graph property. In general, additional effort is required to infer information about global graph properties from local ones [38, Sec. 3.3], [39].

### 3 Random Graphs and Branching Processes

In this section, we review the necessary background related to the proof of Theorem 1 in Section 4.

#### 3.1 Random Graphs

Let  $\mathcal{G}(n, p)$  be the Erdős–Rényi model (also known as the Gilbert model) of a random graph with  $n$  vertices, where each of the  $m = \binom{n}{2}$  possible edges appears with probability  $p$ , independently of all other edges [35, 36]. A helpful representation of this model is to consider a random, symmetric  $n \times n$  adjacency matrix  $\boldsymbol{\theta}$  with entries  $\theta_{i,i} = 0$  and  $\theta_{i,j} (= \theta_{j,i}) \sim \mathbf{B}(p)$ . We use  $G$  to denote a random graph drawn from  $\mathcal{G}(n, p)$ . For the remainder of the paper, we fix  $p = c/n$ .

*Example 4.* Let  $D_k = \sum_{j=1}^n \theta_{k,j}$  be the degree of the  $k$ -th vertex. For any  $k \in [n]$ ,  $D_k \sim \text{Bin}(n-1, c/n)$  with  $\mathbb{E}[D_k] = (n-1)c/n$ . For large  $n$ , all degrees are approximately Poisson distributed with mean  $c$ . More precisely, let  $(D_n)_{n \geq 1}$  be a sequence of RVs denoting the degrees of randomly chosen vertices in  $\mathcal{G}(n, c/n)$  and  $D \sim \text{Po}(c)$ . Then,  $D_n \xrightarrow{d} D$ .  $\triangle$

The following result about the maximum vertex degree will be used in the proof of the concentration bound (D.5).

**Lemma 1.** *Let  $D_{\max} \triangleq \max_{i \in [n]} \sum_{j=1}^n \theta_{i,j}$  be the maximum degree of all vertices in the random graph  $G$ . We have*

$$\mathbb{P}(D_{\max} \geq d_n) \leq e^{-\Omega(d_n)}, \tag{D.6}$$

where  $d_n$  is any function of  $n$  satisfying  $d_n = \Omega(\log(n))$ .

*Proof.* The proof is standard and relies on Chernoff’s inequality and the union bound. For completeness, a proof is given in Appendix A.  $\square$

The random graph  $G$  is completely specified by all its edges, i.e., by the  $m$  RVs  $\theta_{i,j}$  for  $1 \leq j < i \leq n$ . It is sometimes more convenient to specify these RVs in a length- $m$  vector instead of a matrix. With some abuse of notation, we also write  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^\top$ , asserting that there is a one-to-one correspondence between  $\theta_k$  and  $\theta_{i,j}$ .

*Example 5.* Let  $E = \sum_{k=1}^m \theta_k$  be the number of edges in  $G$ . Then,  $E \sim \text{Bin}(m, c/n)$  and the expected number of edges grows linearly with  $n$  since  $\mathbb{E}[E] = mp = (n-1)c/2$ .  $\triangle$

### 3.2 Neighborhood Exploration Process

An important tool to study the neighborhood of a vertex in  $\mathcal{G}(n, p)$  is the so-called exploration process which we briefly review in the following (see, e.g., [40, Sec. 10.4], [41, Ch. 4] for details). This process explores the neighborhood in a breadth-first manner, exposing one vertex at a time. Since we are only interested in exploring the neighborhood up to a fixed depth, we modify the exploration compared to [40, Sec. 10.4], [41, Ch. 4] and stop the process once all vertices in the entire neighborhood for a given depth  $\ell$  are exposed. During the exploration, a vertex can either be active, explored, or neutral. At the beginning (time  $t = 0$ ), one vertex  $v$  is active and the remaining  $n - 1$  vertices are neutral. At each time  $t \geq 1$ , we repeat the following steps.

1. Choose an active vertex that is closest to  $v$  (at time  $t = 1$ , choose  $v$  itself) and denote it by  $w$ .
2. Explore all edges  $(w, w')$ , where  $w'$  runs through all active vertices. If such an edge exists, the explored neighborhood is not a tree. (Apart from this fact, this step has no consequences for the exploration process.)
3. Explore all edges  $(w, w')$ , where  $w'$  runs through all neutral vertices. Set  $w'$  active if the edge exists.
4. Set  $w$  explored.

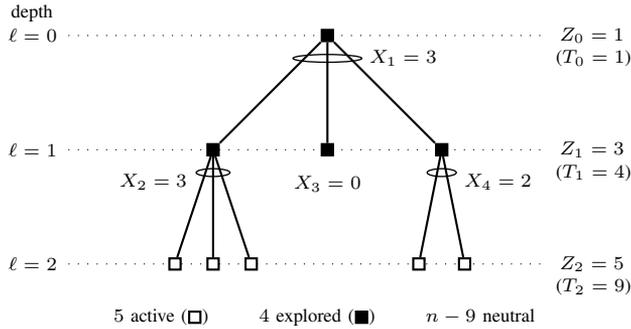
Let  $X_t$  be the number of vertices that become active at time  $t$  (i.e., in step 3). The number of active vertices,  $A_t$ , and neutral vertices,  $N_t$ , at the end of time  $t$  is given by

$$A_t = A_{t-1} + X_t - 1, \quad N_t = n - t - A_t, \tag{D.7}$$

with  $A_0 = 1$ . One can also explicitly write

$$A_t = S_t - (t - 1), \quad N_t = (n - 1) - S_t, \tag{D.8}$$

where  $S_t \triangleq \sum_{i=1}^t X_i$ . Given  $N_{t-1}$ , we have that  $X_t \sim \text{Bin}(N_{t-1}, p)$  because each neutral vertex can become active at time  $t$  with probability  $p$  [40, p.165].



**Figure 3:** The neighborhood of depth  $\ell = 2$  after  $J_2 = 4$  steps in the exploration corresponding to Example 6 in the text.

We define the stopping time  $J_\ell$  of the process  $(X_t)_{t \geq 1}$  to be the time when the entire depth- $\ell$  neighborhood has been exposed.<sup>4</sup> Formally,  $J_\ell$  is recursively defined as

$$J_\ell = \sum_{i=1}^{J_{\ell-1}} X_i + 1 = S_{J_{\ell-1}} + 1, \quad (\text{D.9})$$

for  $\ell \in \mathbb{N}$ , where  $J_0 = 0$  (i.e.,  $J_1 = 1$ ,  $J_2 = X_1 + 1$ ,  $J_3 = \sum_{i=1}^{X_1+1} X_i + 1$ , and so on).

We further use  $Z_\ell$  to denote the number of vertices at depth  $\ell$ , where  $Z_0 = 1$ , and we let  $T_\ell = \sum_{l=0}^{\ell} Z_l$  be the total number of vertices in the entire depth- $\ell$  neighborhood. Observe that  $Z_\ell = A_{J_\ell}$ , i.e., the number of vertices at depth  $\ell$  corresponds to the number of active vertices at the stopping time  $J_\ell$ . We also have  $J_\ell = T_{\ell-1}$ , i.e., the stopping time for depth  $\ell$  corresponds to the number of all vertices up to depth  $\ell - 1$ .

*Example 6.* Assume  $\ell = 2$ . An example of the neighborhood is shown in Fig. 3. The corresponding realization of the stopped exploration process  $(X_1, \dots, X_{J_\ell})$  is given by  $(3, 3, 0, 2)$ , where we assumed a left-to-right ordering of vertices. We have  $J_2 = T_1 = 4$ . Observe that all vertices in the neighborhood are exposed. However, there may still be connections between any of the (active) vertices at depth 2, in which case the neighborhood contains cycles.  $\triangle$

<sup>4</sup>In [40, Sec. 10.4], [41, Ch. 4], the exploration process is used to study the connected components in  $\mathcal{G}(n, p)$ . In that case, the stopping time is commonly defined as the hitting time  $J \triangleq \inf\{t \in \mathbb{N} : A_t = 0\}$ , i.e., the time when we run out of active vertices during the exploration.

### 3.3 Branching Processes

A (Galton–Watson) branching process with offspring distribution  $\bar{\xi}$  is a discrete-time Markov chain  $(\bar{Z}_\ell)_{\ell \geq 0}$  defined by [42, Ch. 8]

$$\bar{Z}_0 = 1 \quad \text{and} \quad \bar{Z}_{\ell+1} = \sum_{i=1}^{\bar{Z}_\ell} \bar{\xi}_{\ell,i}, \quad (\text{D.10})$$

where  $(\bar{\xi}_{\ell,i})_{\ell,i \geq 0}$  is a two-dimensional sequence of i.i.d.  $\mathbb{N}_0$ -valued RVs with distribution  $\bar{\xi}_{\ell,i} \sim \bar{\xi}$ . In our context, the interpretation of the process is as follows. Start with one vertex at depth  $\ell = 0$  which has a random number of neighboring (or offspring) vertices extending to depth 1. Each of the vertices at depth 1 (if there are any) has again a random number of offspring vertices, independently of all other vertices, and so on.  $\bar{Z}_\ell$  is the total number of vertices at depth  $\ell$ , whereas  $\bar{\xi}_{\ell,i}$  is the number of offspring vertices of the  $i$ th vertex at depth  $\ell$ . We further define the total number of vertices up to (and including) depth  $\ell$  as  $\bar{T}_\ell = \sum_{l=0}^{\ell} \bar{Z}_l$ .

The exploration process in the previous subsection is closely related to a Poisson branching process with mean  $c$ , i.e., the case where  $\bar{\xi} = \text{Po}(c)$ . The connection becomes apparent by considering the random-walk perspective of the branching process [41, Sec. 3.3]. Here, the number of offspring vertices is specified in a one-dimensional fashion, indexed by  $t$ , and denoted by  $\bar{X}_t$ . The indexing is done breadth-first, in a predetermined order, e.g., left to right. In particular, we have

$$\bar{A}_t = \bar{A}_{t-1} + \bar{X}_t - 1 \quad (\text{D.11})$$

with  $\bar{A}_0 = 1$ , similar to (D.7). The crucial difference with respect to the exploration process is that  $\bar{X}_t \sim \bar{\xi}$  for all  $t$ .

Similar to the exploration process, we recursively define the stopping time for the process  $(\bar{X}_t)_{t \geq 1}$  as  $\bar{J}_\ell = \sum_{i=1}^{\bar{J}_{\ell-1}} \bar{X}_i + 1$  with  $\bar{J}_0 = 0$  (cf. (D.9)), where  $\bar{J}_\ell = \bar{T}_{\ell-1}$ . Thus, the stopped process  $(\bar{X}_1, \dots, \bar{X}_{\bar{J}_\ell})$  specifies the branching process up to depth  $\ell$ .

## 4 Proof of Theorem 1

In the following, we provide a proof of Theorem 1. In Section 4.1, we show that, with high probability, the depth- $\ell$  neighborhood of a vertex in the residual graph  $G$  is a tree. We use this result in Section 4.2 to show the convergence of the expected decoding outcome for an individual CN after  $\ell$  iterations to the decoding outcome when evaluated on the branching process. The iterative decoding on the branching process (also known as DE) is analyzed in Section 4.3. Finally, the concentration bound in (D.5) is shown in Section 4.4.

The tree-like behavior and the convergence of the neighborhood in  $\mathcal{G}(n, c/n)$  to the Poisson branching process are certainly well-known within the random-graph-theory literature. For example, this type of convergence is sometimes referred to as local weak

convergence, see, e.g., [43] or [44, Prop. 2.3.1]. Here, we give a simple proof based on stochastic processes and stopping times.

## 4.1 Tree-like Neighborhood

**Lemma 2.** *Let  $B_G(k, \ell)$  denote the depth- $\ell$  neighborhood of the  $k$ -th vertex in  $G$ . Then, for any  $k \in [n]$ , we have*

$$\mathbb{P}(B_G(k, \ell) \text{ is a tree}) \geq 1 - \frac{\beta(c, \ell)}{n}, \quad (\text{D.12})$$

where  $\beta(c, \ell)$  depends only on  $c$  and  $\ell$ .

*Proof.* We can use the exploration process in Section 3.2 to show that the total number of potential edges that could create a cycle during the exploration (i.e., in step 2) is given by

$$N_\ell = \sum_{i=1}^{J_\ell} (A_{i-1} - 1) + \binom{Z_\ell}{2}. \quad (\text{D.13})$$

In particular, at each time  $t$  up to the random stopping time  $J_\ell$ , there exist  $A_{t-1} - 1$  “exposed” vertices, in the sense that these vertices are known to be part of the neighborhood, but have not yet been explored. Each of the  $A_{t-1} - 1$  potential edges to these vertices creates a cycle. Furthermore, at the stopping time  $J_\ell$ , there exist  $Z_\ell$  exposed vertices at depth  $\ell$ , with  $\binom{Z_\ell}{2}$  potential edges between them, each of which creates a cycle (see, e.g., Fig. 3). For the neighborhood to be a tree, all of these edges must be absent. Since any edge in the exploration will not appear with probability  $1 - c/n$ , independently of all other edges, we have

$$\mathbb{P}(B_G(k, \ell) \text{ is a tree}) = \mathbb{E} \left[ \left(1 - \frac{c}{n}\right)^{N_\ell} \right] \quad (\text{D.14})$$

$$\geq 1 - \frac{c}{n} \mathbb{E}[N_\ell]. \quad (\text{D.15})$$

Surely,  $N_\ell$  cannot be larger than the total number of possible edges in the neighborhood, i.e.,  $N_\ell \leq \binom{T_\ell}{2} \leq T_\ell^2/2$ , where we recall that  $T_\ell$  is the total number of vertices encountered. Inserting this bound into (D.15) and using the bound (D.106) on  $\mathbb{E}[T_\ell^2]$  in Appendix B (which depends only on  $c$  and  $\ell$ ) completes the proof.  $\square$

*Remark 6.* The analogous result for (regular) LDPC code ensembles is given in [17, App. A] (see also [45, Sec. 2.2]). The main difference with respect to the proof in [17, App. A] (and its extension to irregular ensembles with bounded maximum VN and CN degree) is that the number of vertices in the neighborhood cannot be upper bounded by a constant which is independent of  $n$ . (In [17],  $n$  corresponds to the LDPC code length.)

## 4.2 Convergence to the Poisson Branching Process

It is well-known that the degree of a vertex in  $\mathcal{G}(n, c/n)$  converges to a Poisson RV with mean  $c$  as  $n \rightarrow \infty$  (see Example 4). More generally, for any finite  $t$ , and any  $(x_1, \dots, x_t) \in \mathbb{N}_0^t$ , one can easily show that (see, e.g., [41, Sec. 4.1.2])

$$\lim_{n \rightarrow \infty} f_{X_1, \dots, X_t}(x_1, \dots, x_t) = f_{\bar{X}_1}(x_1) \cdot \dots \cdot f_{\bar{X}_t}(x_t), \quad (\text{D.16})$$

where  $\bar{X}_1, \dots, \bar{X}_t$  are i.i.d.  $\text{Po}(c)$ . This, together with Lemma 2, implies that the distribution on the shape of the neighborhood (for any fixed depth) converges to a Poisson branching process with mean  $c$ . To see this, note that under the assumption that the neighborhood is tree-like, its shape is specified by the stopped exploration process  $(X_1, \dots, X_{J_\ell})$ . Each realization of  $(X_1, \dots, X_{J_\ell})$  is a vector of some (finite) length specifying the number of offspring vertices in the tree in a sequential manner. The set of all realizations is thus a subset of  $\mathbb{N}_0^* = \mathbb{N}_0 \cup \mathbb{N}_0^2 \cup \mathbb{N}_0^3 \cup \dots$ . Since  $\mathbb{N}_0^*$  is countably infinite, there exists a one-to-one mapping between  $\mathbb{N}_0^*$  and  $\mathbb{N}_0$ . We denote such a mapping by  $\mathcal{M} : \mathbb{N}_0^* \rightarrow \mathbb{N}_0$  and let  $\mathcal{M}^{-1}$  be its inverse. We now define new RVs  $B_n = \mathcal{M}(X_1, \dots, X_{J_\ell})$  and  $B = \mathcal{M}(\bar{X}_1, \dots, \bar{X}_{\bar{J}_\ell})$ . One can think about enumerating all possible trees and assigning an index to each of them. A distribution over the shape of the trees is then equivalent to a distribution over the indices. It is now easy to show that  $B_n \xrightarrow{d} B$ . For any  $b \in \mathbb{N}_0$ , there exists some  $t$  such that  $\mathcal{M}^{-1}(b) = (x_1, \dots, x_t) \in \mathbb{N}_0^t$ . Therefore, we have

$$\lim_{n \rightarrow \infty} \mathbb{P}(B_n = b) \quad (\text{D.17})$$

$$= \lim_{n \rightarrow \infty} f_{J_\ell | X_1, \dots, X_t}(t | x_1, \dots, x_t) f_{X_1, \dots, X_t}(x_1, \dots, x_t) \quad (\text{D.18})$$

$$= f_{\bar{J}_\ell | \bar{X}_1, \dots, \bar{X}_t}(t | x_1, \dots, x_t) \lim_{n \rightarrow \infty} f_{X_1, \dots, X_t}(x_1, \dots, x_t) \quad (\text{D.19})$$

$$\stackrel{(\text{D.16})}{=} f_{\bar{J}_\ell | \bar{X}_1, \dots, \bar{X}_t}(t | x_1, \dots, x_t) f_{\bar{X}_1}(x_1) \cdot \dots \cdot f_{\bar{X}_t}(x_t) \quad (\text{D.20})$$

$$= \mathbb{P}(B = b), \quad (\text{D.21})$$

where, to obtain (D.19) from (D.18), we used the fact that the conditional distributions of the stopping times  $J_\ell$  and  $\bar{J}_\ell$  given  $X_1, \dots, X_t$  and  $\bar{X}_1, \dots, \bar{X}_t$ , respectively, are equal and independent of  $n$ . These distributions are simply indicator functions for the event that  $(x_1, \dots, x_t) \in \mathbb{N}_0^t$  fully specifies a tree of depth  $\ell$ .

A direct consequence of this result is that the expected value of a (bounded) function applied to the neighborhood of a vertex in  $\mathcal{G}(n, c/n)$  converges to the expected value of the same function applied to the branching process. In particular, recall that the RV  $W = \frac{1}{n} \sum_{k=1}^n W_k$  corresponds to the fraction of component codes that declare failures after  $\ell$  decoding iterations. The indicator RV  $W_k$  depends only on the shape of the depth- $\ell$  neighborhood of the  $k$ -th vertex in the residual graph. The peeling procedure can thus be written using a function  $\mathcal{D}_\ell : \mathbb{N}_0 \rightarrow \{0, 1\}$ , such that

$$\mathbb{E}[W_k | B_G(k, \ell) \text{ is a tree}] = \mathbb{E}[\mathcal{D}_\ell(B_n)], \quad (\text{D.22})$$

which, due to symmetry, is independent of  $k$ . Since  $B_n \xrightarrow{d} B$  and  $\mathcal{D}_\ell$  is bounded, we have that [46, Sec. 10]

$$\lim_{n \rightarrow \infty} \mathbb{E}[\mathcal{D}_\ell(B_n)] = \mathbb{E}[\mathcal{D}_\ell(B)] = z^{(\ell)}, \quad (\text{D.23})$$

which, together with (D.12), implies (D.4).

*Remark 7.* It is worth mentioning that for regular LDPC code ensembles, there is no notion of an asymptotic neighborhood distribution (in the sense of (D.16)) beyond the fact that cycles can be ignored. This is because the ensemble of computation trees for a CN (or VN) reduces to a single deterministic tree.

### 4.3 Density Evolution

Once the true distribution on the neighborhood-shape has been replaced by the branching process, the parameter of interest can be easily computed (cf. [34, Sec. 2], [38, p. 43]). In our case, the parameter of interest is the probability that a CN declares a decoding failure after  $\ell$  iterations as  $n \rightarrow \infty$ , or, equivalently, the probability that the root vertex of the branching process survives  $\ell$  peeling iterations. Due to the recursion that is inherent in the definition of the branching process, it is not surprising that the solution is also given in terms of a recursion. This is, of course, completely analogous to the analysis of LDPC code ensembles, see, e.g., the discussion in [45, Sec. 1]. Also, similar to LDPC codes over the BEC, we refer to this step as DE (even though the parameter of interest does not correspond to a density).

Consider a Poisson branching process with mean  $c$ . Assume that we have a realization of this process (i.e., a tree) up to depth  $\ell$ . We wish to determine if the root vertex survives  $\ell$  iterations of the peeling procedure (and thus the CN corresponding to the root node declares a decoding failure). One can recursively break down the answer as follows. First, for each of the root's offspring vertices, apply  $\ell - 1$  peeling iterations to the subtree that has the offspring vertex as a root (and extends from depth 1 to  $\ell$ ). Then, if the number of offspring vertices that survive this peeling is less than or equal to  $\mathfrak{t}$ , remove the root vertex. This gives the same answer as applying  $\ell$  peeling iterations to the entire tree, since we are simply postponing the removal decision for the root to the  $\ell$ -th iteration.

Now, in order to determine the corresponding *probability* with which the root vertex survives the peeling procedure, the crucial observation is that the root's offspring vertices are removed independently of each other, and with the same probability. This is a simple consequence of the definition of the branching process and the independence assumption between the number of offspring vertices (see Section 3.3). Recall that we defined the root survival probability as  $z^{(\ell)}$ . Furthermore, we denote the survival probability of the root's offspring vertices by  $x^{(\ell-1)}$ . Initially, the number of offspring vertices is Poisson distributed with mean  $c$ . After removing each offspring vertex independently with probability  $1 - x^{(\ell-1)}$ , the offspring distribution of the root vertex follows again a

Poisson distribution, albeit with (reduced) mean  $cx^{(\ell-1)}$ . (This is easily seen by using characteristic functions.) Hence, we obtain (D.2).

Essentially the same argument can be used to determine  $x^{(\ell)}$ . The only difference is that for offspring vertices we have to account for the fact they are connected to the previous level with an edge. Thus, they can be removed only if less than or equal to  $\mathfrak{t} - 1$  (and not  $\mathfrak{t}$ ) of their offspring vertices survive. This leads to the recursion (D.3), where the initial condition is given by  $x^{(0)} = 1$ .

## 4.4 Concentration

The concentration bound in (D.5) is readily proved by using the method of typical bounded differences [47]. In particular, we can apply a special case of [47, Cor. 1.4] which is stated below (with adjusted notation) for easier referencing.

**Theorem 2** ([47]). *Let  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^\top$  be a vector of independent RVs with  $\theta_k \sim \mathbb{B}(p)$  for all  $k$ . Let  $\Gamma \subseteq \{0, 1\}^m$  be an event and let  $f : \{0, 1\}^m \rightarrow \mathbb{R}$  be a function that satisfies the following condition. There exist  $\Lambda$  and  $\Lambda'$  with  $\Lambda \leq \Lambda'$  such that whenever  $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \{0, 1\}^m$  differ in only one coordinate, we have*

$$|f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}')| \leq \begin{cases} \Lambda & \text{if } \boldsymbol{\theta} \in \Gamma \\ \Lambda' & \text{otherwise} \end{cases}. \quad (\text{D.24})$$

Then, for any  $a \geq 0$  and any choice of  $\gamma \in (0, 1]$ , we have

$$\begin{aligned} \mathbb{P}(|f(\boldsymbol{\theta}) - \mathbb{E}[f(\boldsymbol{\theta})]| \geq a) &\leq m\gamma^{-1}\mathbb{P}(\boldsymbol{\theta} \notin \Gamma) \\ &+ \exp\left(-\frac{a^2}{2m(1-p)p(\Lambda+b)^2 + 2(\Lambda+b)a/3}\right), \end{aligned} \quad (\text{D.25})$$

where  $b = \gamma(\Lambda' - \Lambda)$ .

In our context,  $\boldsymbol{\theta}$  specifies the edges in the random graph  $G$  (see Section 3). Thus, we can think about  $\boldsymbol{\theta}$  and  $\boldsymbol{\theta}'$  as specifying two different graphs  $G = G(\boldsymbol{\theta})$  and  $G' = G(\boldsymbol{\theta}')$ . The interpretation of the condition (D.24) is as follows. For any two graphs  $G, G'$  that differ in only one edge, we have  $|f(G) - f(G')| \leq \Lambda'$ , where  $f$  denotes a function applied to the graphs. The constant  $\Lambda'$  is often referred to as the Lipschitz constant [47]. The event  $\Gamma$  is chosen such that changing one coordinate in  $\boldsymbol{\theta} \in \Gamma$  (i.e., adding or removing an edge in the graph defined by  $\boldsymbol{\theta}$ ) changes the function by at most  $\Lambda$ , where  $\Lambda$  should be substantially smaller than  $\Lambda'$ . The constant  $\Lambda$  is referred to as the typical Lipschitz constant. In this regard, the event  $\Gamma$  is assumed to be a typical event, i.e., it should occur with high probability.

*Remark 8.* In several applications, it is possible to establish concentration bounds based solely on suitable choices for  $\Lambda'$ . This approach leads to the more common bounded differences inequality (also known as McDiarmid's or Hoeffding-Azuma inequality). For

example, the concentration bound for LDPC code ensembles in [17, Eq. (11)] is based on this approach. However, in many cases (including the one considered here) the worst case changes corresponding to  $\Lambda'$  can be quite large, even though the typical changes may be small. For more details, we refer the reader to [47] and references therein.

Theorem 2 is applied as follows. We let  $f(\boldsymbol{\theta}) = nW = \sum_{k=1}^n W_k$ . Since  $f$  is the sum of  $n$  indicator RVs, we can choose  $\Lambda' = n$ . We further let  $\Gamma$  be the event that the maximum vertex degree in  $G$ , denoted by  $D_{\max}$ , is strictly less than  $n^\delta$  for some fixed  $\delta \in (0, 1)$ . For the typical Lipschitz constant, we choose  $\Lambda = 2(\ell + 1)n^{\delta\ell}$ . To show that for these choices the condition (D.24) holds, we argue as follows. First, observe that the maximum vertex degree in both  $G$  and  $G'$  is at most  $n^\delta$  since adding an edge to the graph  $G$  increases the maximum degree by at most one (and removing an edge can only decrease the maximum degree). Consider now the maximum change in  $\sum_{k=1}^n W_k$  that can occur by adding or removing an edge between two arbitrary vertices  $i$  and  $j$  under the assumption that the maximum degree remains bounded by  $n^\delta$ . Since  $W_k$  depends only on the depth- $\ell$  neighborhood of the  $k$ -th vertex, such a change can only affect  $W_k$  if either vertex  $i$  or  $j$  (or both) are part of the neighborhood of vertex  $k$ . But, due to the bounded maximum degree, vertex  $i$  appears in at most  $\sum_{l=0}^{\ell} n^{\delta l} \leq (\ell + 1)n^{\delta\ell}$  neighborhoods (and so does vertex  $j$ ). Hence, the sum  $\sum_{k=1}^n W_k$  can change by at most  $2(\ell + 1)n^{\delta\ell}$ .

We further choose  $\gamma = n^{-1}$ . Since  $\Lambda' = n$ , this implies that  $b \leq \gamma\Lambda' = 1$  and therefore we have

$$(\Lambda + b) \leq (\Lambda + b)^2 \leq 4\Lambda^2. \quad (\text{D.26})$$

Consider now the second term on the right-hand side (RHS) of (D.25) with  $a = n\varepsilon$  and  $p = c/n$ . We have

$$\exp\left(\frac{-(n\varepsilon)^2}{2m(1 - c/n)c/n(\Lambda + b)^2 + 2(\Lambda + b)n\varepsilon/3}\right) \quad (\text{D.27})$$

$$\leq \exp\left(\frac{-\varepsilon^2 n}{(8c + 8\varepsilon/3)\Lambda^2}\right) = e^{-\beta_1 n^{1-2\delta\ell}} \quad (\text{D.28})$$

where the inequality in (D.28) follows from  $m \leq n^2$ ,  $1 - c/n \leq 1$  and (D.26), and in the last step we used  $\Lambda = 2(\ell + 1)n^{\delta\ell}$ . Note that the implicitly defined parameter  $\beta_1 > 0$  depends only on  $\varepsilon$ ,  $c$ , and  $\ell$ . In order to bound the first term on the RHS of (D.25), we first note that  $\mathbb{P}(\boldsymbol{\theta} \notin \Gamma) = \mathbb{P}(D_{\max} \geq n^\delta)$ . We then have

$$m\gamma^{-1}\mathbb{P}(\boldsymbol{\theta} \notin \Gamma) \leq n^3 e^{-\beta_2 n^\delta} \leq e^{-\beta_2 n^{\delta/2}}, \quad (\text{D.29})$$

where, according to Lemma 1, the first inequality holds for some  $\beta_2 > 0$  and  $n$  sufficiently large. To match the exponents in (D.28) and (D.29), we can set  $\delta = (1 + 2\ell)^{-1}$ . This proves (D.5) and completes the proof of Theorem 1.

*Remark 9.* The above proof applies to any function of the form  $f = \sum_{k=1}^n f_k$  where  $f_k$  is an indicator function that depends only on the depth- $\ell$  neighborhood of the  $k$ -th vertex in  $G$ .

## 5 Generalized Product Codes

In this section, we analyze a deterministic construction of GPCs for which the residual graph corresponds to an inhomogeneous random graph [23]. The concept of inhomogeneity naturally arises if we wish to distinguish between different types of vertices. In our case, a type will correspond to a particular position in the Tanner graph and a certain erasure-correcting capability. HPCs can be regarded as “single-type” or homogeneous, in the sense that all CNs (and thus all vertices in the residual graph) behave essentially the same.

### 5.1 Code Construction

Our code construction is defined in terms of three parameters  $\boldsymbol{\eta}$ ,  $\boldsymbol{\gamma}$ , and  $\boldsymbol{\tau}$ . We denote the corresponding GPC by  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$ , where  $n$  denotes the total number of CNs in the underlying Tanner graph. The two parameters  $\boldsymbol{\eta}$  and  $\boldsymbol{\gamma}$  essentially determine the graph connectivity, where  $\boldsymbol{\eta}$  is a binary, symmetric  $L \times L$  matrix and  $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_L)^\top$  is a probability vector of length  $L$ , i.e.,  $\sum_{i=1}^L \gamma_i = 1$  and  $\gamma_i \geq 0$ . Since GPCs have a natural representation in terms of two-dimensional code arrays (see, e.g., Fig. 5), one may alternatively think about  $\boldsymbol{\eta}$  and  $\boldsymbol{\gamma}$  as specifying the array shape. We will see in the following that different choices for  $\boldsymbol{\eta}$  and  $\boldsymbol{\gamma}$  recover well-known code classes. The parameter  $\boldsymbol{\tau}$  is used to specify GPCs employing component codes with different erasure-correcting capabilities and will be described in more detail at the end of this subsection.

The Tanner graph describing the GPC  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  is constructed as follows. Assume that there are  $L$  positions. Place  $n_i \triangleq \gamma_i n$  CNs at each position  $i \in [L]$ , where we assume that  $n_i$  is an integer for all  $i$ . Then, connect each CN at position  $i$  to each CN at position  $j$  through a VN if and only if  $\eta_{i,j} = 1$ .

In the following, we always assume that  $\eta_{i,j} = 1$  for at least one  $j$  and any  $i \in [L]$  so that there are no unconnected CNs. Furthermore, we assume that the matrix  $\boldsymbol{\eta}$  is irreducible, so that the Tanner graph is not composed of two (or more) disconnected graphs.

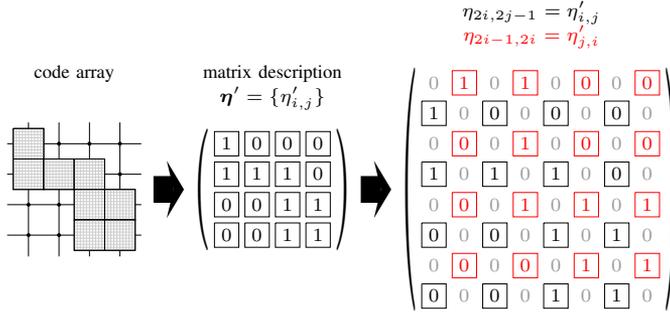
Each of the  $n_i$  CNs at position  $i$  has degree

$$d_i = \eta_{i,i}(n_i - 1) + \sum_{j \neq i} \eta_{i,j} n_j. \quad (\text{D.30})$$

Thus,  $d_i$  is the length of the component codes associated with CNs at position  $i$ . The first term in (D.30) arises from the fact that we cannot connect a CN to itself if  $\eta_{i,i} = 1$ . The total number of VNs (i.e., the length of the code) is given by

$$m = \sum_{i=1}^L \eta_{i,i} \binom{n_i}{2} + \sum_{1 \leq i < j \leq L} \eta_{i,j} n_i n_j \approx \frac{\boldsymbol{\gamma}^\top \boldsymbol{\eta} \boldsymbol{\gamma}}{2} n^2. \quad (\text{D.31})$$

In the following, we assume some fixed (and arbitrary) ordering on the CNs and VNs.



**Figure 4:** Construction of  $\eta$  for an arbitrary code array composed of blocks that are arranged on a grid. Red elements in  $\eta$  are inserted such that  $\eta$  is symmetric. With this construction, even (odd) positions in  $\eta$  correspond to row (column) codes.

*Remark 10.* In the light of Remark 1, we see that the above construction merely specifies a Tanner graph and not a code. This is due to the missing assignment of the component code bit positions to the CN edges. Since our results do not depend on this assignment, it is assumed to be (arbitrarily) fixed. In the following examples, the assignment is implicitly specified due to an array description.

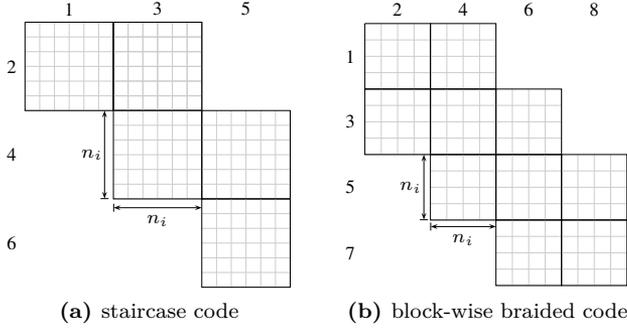
*Example 7.* HPCs are recovered by considering  $\eta = 1$  and  $\gamma = 1$ . All CNs are equivalent and correspond to component codes of length  $n - 1$ .  $\triangle$

*Example 8.* Choosing  $\eta = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$  leads to a PC. The relative lengths of the row and column component codes can be adjusted through  $\gamma$ , where  $\gamma = (1/2, 1/2)$  leads to a “square” PC with (uniform) component code length  $n/2$ . Note that the total number of CNs  $n$  is assumed to be even in this case.  $\triangle$

*Example 9.* Consider an arbitrarily shaped code array of finite size which is composed of blocks of size  $n' \times n'$  arranged on a square grid. In Fig. 4, we illustrate how to construct  $\eta$  for such an array. First, form the matrix  $\eta'$  representing the array, where entries are 1 if a block is present on the corresponding grid point and 0 otherwise. Assuming that  $\eta'$  has size  $a \times b$ , the matrix  $\eta$  of size  $(a + b) \times (a + b)$  is then constructed by using the prescription

$$\begin{aligned} \eta_{2i,2j-1} &= \eta'_{i,j}, \\ \eta_{2i-1,2j} &= \eta'_{j,i} \end{aligned} \tag{D.32}$$

for  $i \in [a]$  and  $j \in [b]$  and  $\eta_{i,j} = 0$  elsewhere. For example, consider a PC where  $\gamma = (1/3, 2/3)$ , i.e., the relative length of the row and column codes is 2 : 3. An alternative way of describing the code is to consider a matrix description of the array



**Figure 5:** Examples of code arrays for (a) staircase codes (see Example 10) and (b) block-wise braided codes (see Example 11).

according to

$$\boldsymbol{\eta}' = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad (\text{D.33})$$

and then form  $\boldsymbol{\eta}$  according to (D.32). The size of each block is assumed to be  $n' = n/5$ . For the alternative description we have  $L = 5$  and  $\gamma_i = 1/5$  for all  $i$ .  $\triangle$

*Example 10.* For a fixed  $L \geq 2$ , the matrix  $\boldsymbol{\eta}$  describing a staircase code [8] has entries  $\eta_{i,i+1} = \eta_{i+1,i} = 1$  for  $i \in [L - 1]$  and zeros elsewhere. The distribution  $\boldsymbol{\gamma}$  is uniform, i.e.,  $\gamma_i = 1/L$  for all  $i \in [L]$ . For example, the staircase code corresponding to the code array shown in Fig. 5(a), where  $L = 6$  and  $n = 36$  (i.e.,  $n_i = 6$ ), is defined by

$$\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (\text{D.34})$$

and  $\gamma_i = 1/6$ . The CNs at all positions have the same degree  $2n\gamma_i = 12$ , except for positions 1 and  $L$ , where the degrees are  $n\gamma_i = 6$ .  $\triangle$

*Example 11.* For even  $L \geq 4$ , the matrix  $\boldsymbol{\eta}$  for a particular instance of a block-wise braided code has entries  $\eta_{i,i+1} = \eta_{i+1,i} = 1$  for  $i \in [L - 1]$ ,  $\eta_{2i-1,2i+2} = \eta_{2i+2,2i-1} = 1$

for  $i \in [L/2 - 1]$ , and zeros elsewhere. For example, we have

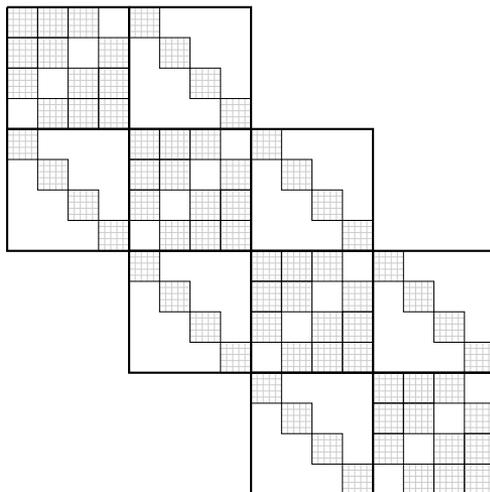
$$\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (\text{D.35})$$

for  $L = 8$ . The corresponding code array is shown in Fig. 5(b), where  $n = 32$  and  $\gamma$  is uniform. In general, the construction of a block-wise braided code is based on so-called multiple block permutators (MBPs). An MBP with multiplicity  $k$  is an  $N \times N$  matrix with  $k$  ones in each row and column [22, Def. 2.1]. Given a component code of length  $n_C$  and dimension  $k_C$ , the diagonal and off-diagonal array blocks in Fig. 5(b) correspond to MBPs with respective multiplicities  $2k_C - n_C$  and  $n_C - k_C$ , where  $N \geq \min(2k_C - n_C, n_C - k_C)$ . However, this definition is unnecessarily narrow for our purposes in the sense that the multiplicities of the MBPs are linked to the dimension of the component code. For example, for the array shown in Fig. 5(b) (where  $N = 4$  and  $n = 12$ ), it would be required that each component code has dimension  $k_C = 8$  in order to comply with the definition in [22]. Here, we simply lift the constraint that the multiplicities of the MBPs are linked to the component code dimension. The only requirement for the considered GPC construction is that the MBPs have a block-wise structure themselves, see Fig. 6 for an example. Note that  $\boldsymbol{\eta}$  can be found by following the steps in Example 9.  $\triangle$

*Remark 11.* Both staircase and braided codes were originally introduced as convolutional-like codes with conceptually infinite length, i.e.,  $L = \infty$ . It then becomes customary to employ a sliding-window decoder whose analysis is discussed in Section 6.3. We also remark that it is straightforward to extend the above construction of  $\boldsymbol{\eta}$  and  $\boldsymbol{\gamma}$  for staircase and braided codes to their natural tail-biting versions (see, e.g., [11]).

Up to this point, the GPC construction for a given  $\boldsymbol{\eta}$ ,  $\boldsymbol{\gamma}$ , and  $n$  specifies the lengths of the component codes via (D.30). We proceed by assigning different erasure-correcting capabilities to the component codes corresponding to CNs at different positions. To that end, for  $i \in [L]$ , let  $\boldsymbol{\tau}(i) = (\tau_1(i), \dots, \tau_{\mathbf{t}_{\max}}(i))^\top$  be a probability vector of length  $\mathbf{t}_{\max}$ , where  $\tau_{\mathbf{t}}(i)$  denotes the fraction of CNs at position  $i$  (out of  $n_i$  total CNs) which can correct  $\mathbf{t}$  erasures and  $\mathbf{t}_{\max}$  is the maximum erasure-correcting capability. With some abuse of notation, the collection of these distributions for all positions is denoted by  $\boldsymbol{\tau} = (\boldsymbol{\tau}(i))_{i=1}^L$ . The assignment can be done either deterministically, assuming that  $\tau_{\mathbf{t}}(i)n_i$  is an integer for all  $i \in [L]$  and  $\mathbf{t} \in [\mathbf{t}_{\max}]$ , or independently at random according to the distribution  $\boldsymbol{\tau}(i)$  for each position.

*Example 12.* Consider a PC where the row and column codes have the same length but different erasure-correcting capabilities  $\mathbf{t}$  and  $\mathbf{t}'$ , respectively. We have  $\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ ,



**Figure 6:** A block-wise braided code where the MBPs have a block-wise structure. We have  $N = 20$ ,  $n = 160$ , and the multiplicities for the diagonal and off-diagonal MBPs are 15 and 5, respectively.

$\gamma = (1/2, 1/2)$ , and additionally  $\tau_t(1) = 1$  and  $\tau_{t'}(2) = 1$ . More generally, the erasure-correcting capabilities may also vary across the row (and column) codes leading to irregular PCs [20, 21].  $\triangle$

*Example 13.* Staircase codes with component code mixtures were suggested (but not further investigated) in [48, Sec. 4.4.1]. The case described in [48, Sec. 4.4.1] corresponds to a fixed choice of  $\tau(i)$  which is independent of  $i$ .  $\triangle$

## 5.2 Inhomogeneous Random Graphs

Assume that a codeword of  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  is transmitted over the BEC with erasure probability  $p = c/n$ , for  $c > 0$ . Recall that the residual graph is obtained by removing known VNs and collapsing erased VNs into edges. We now illustrate how the ensemble of residual graphs for  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  is related to the inhomogeneous random graph model.

In [23], inhomogeneous random graphs are specified by a vertex space  $\mathcal{V}$  and a kernel  $\kappa$ . Here, we consider only the finite-type case, see [23, Ex. 4.3]. In this case, the number of different vertex types is denoted by  $r$  and the vertex space  $\mathcal{V}$  is a triple  $(\mathcal{S}, \mu, (\mathbf{y}_n)_{n \geq 1})$ , where  $\mathcal{S} = [r]$  is the so-called type space,  $\mu : \mathcal{S} \rightarrow [0, 1]$  is a probability measure on  $\mathcal{S}$ , and  $\mathbf{y}_n = (y_1^{(n)}, y_2^{(n)}, \dots, y_n^{(n)})$  is a deterministic or random sequence of points in  $\mathcal{S}$  such that for each  $i \in \mathcal{S}$ , we have

$$\frac{|\{k : y_k^{(n)} = i\}|}{n} \xrightarrow{\mathbb{P}} \mu(i) \quad (\text{D.36})$$

as  $n \rightarrow \infty$ . For a finite number of vertex types, the kernel  $\kappa$  is a symmetric  $r \times r$  matrix. Therefore, we denote the kernel by  $\boldsymbol{\kappa}$  where entries are denoted by  $\kappa_{i,j}$ . For a fixed  $n > \max_{i,j} \kappa_{i,j}$ , the inhomogeneous random graph  $\mathcal{G}^{\mathcal{V}}(n, \boldsymbol{\kappa})$  is defined as follows. The graph has  $n$  vertices where the type of vertex  $i$  is given by  $y_i^{(n)}$ . An edge between vertex  $i$  and  $j$  exists with probability  $n^{-1} \kappa_{x_i^{(n)}, x_j^{(n)}}$ , independently of all other edges.

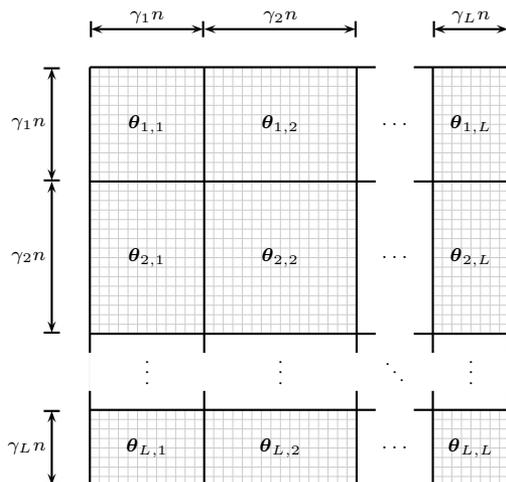
*Remark 12.* Even though we use [23] as our main reference, finite-type inhomogeneous random graphs (and their relation to multi-type branching processes) were first introduced and studied in [49]. See also the discussion in [38, p. 31].

For the code  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$ , the residual graph is an instance of an inhomogeneous random graph with a finite number of types, as defined above. In particular, there are  $r = L t_{\max}$  different types in total, i.e., we have  $\mathcal{S} = [L t_{\max}]$ . In our case, it is more convenient to specify the type of a vertex by a pair  $(i, \mathbf{t})$ , where  $i \in [L]$  corresponds to the position in the Tanner graph and  $\mathbf{t} \in [t_{\max}]$  corresponds to the erasure-correcting capability. In the construction of the sequence  $\mathbf{x}_n$ , the assignment of the type corresponding to the position is always deterministic. For the type corresponding to the erasure-correcting capability, we have the freedom to do the assignment deterministically or uniformly at random. In both cases, the fraction of vertices of type  $(i, \mathbf{t})$  is asymptotically given by  $\gamma_i \tau_{\mathbf{t}}(i)$ . This specifies the probability measure  $\mu$  through the condition (D.36). (For the random assignment, the condition (D.36) holds due to the weak law of large numbers.) The kernel  $\boldsymbol{\kappa}$  is obtained from  $\boldsymbol{\eta}$  by replacing each 0 entry with the all-zero matrix of size  $t_{\max} \times t_{\max}$  and each 1 entry with a  $t_{\max} \times t_{\max}$  matrix where all entries are equal to  $c$ .

*Remark 13.* The inhomogeneous random graph model in [23] is much more general than the finite-type case described above. In particular,  $\mathcal{S}$  can be a separable metric space and  $\kappa$  a symmetric non-negative (Borel) measurable function on  $\mathcal{S} \times \mathcal{S}$ . This more general framework could be used for example to obtain the DE equations for so-called tightly-braided codes [13, 22]. However, in that case the analysis does not admit a characterization in terms of a finite number of types. In particular, the DE equations are given in terms of integrals and solving the equations may then require the application of numerical integration techniques.

Similar to the (homogeneous) random graph  $\mathcal{G}(n, p)$ , one may use an alternative representation in terms of a random, symmetric  $n \times n$  adjacency matrix  $\boldsymbol{\theta}$  with zeros on the diagonal. The structure of this matrix is shown in Fig. 7. The matrix is composed of submatrices  $\boldsymbol{\theta}_{i,j}$  of size  $n_i \times n_j$ . The submatrix  $\boldsymbol{\theta}_{i,j}$  is zero if  $\eta_{i,j} = 0$  and it consists of i.i.d.  $\mathcal{B}(p)$  RVs if  $\eta_{i,j} = 1$  (with the constraint that the matrix  $\boldsymbol{\theta}$  is symmetric and all diagonal elements are zero). The inhomogeneous random graph is thus specified by  $m$  Bernoulli RVs, where  $m$  is defined in (D.31).

From the matrix representation, it can be seen that the degree of a vertex at position  $k$  is distributed according to  $\text{Bin}(d_k, c/n)$ , where  $d_k$  is defined in (D.30). Moreover, all vertices follow a Poisson distribution as  $n \rightarrow \infty$ , where the mean for vertices at position



**Figure 7:** The structure of the random, symmetric adjacency matrix  $\theta$ .

$k$  is given by

$$\lim_{n \rightarrow \infty} d_k \frac{c}{n} = c \sum_{j=1}^L \gamma_j \eta_{k,j}. \quad (\text{D.37})$$

### 5.3 Iterative Decoding

After transmission over the BEC, we apply the same iterative decoding as described in Section 2.3 for the HPC. The only difference is that each component code is assumed to correct all erasures up to its erasure-correcting capability. In the corresponding iterative peeling procedure for the residual graph, one removes vertices of degree at most  $t$ , where  $t$  is the erasure-correcting capability of the corresponding CN. The erasure-correcting capability may now be different for different vertices depending on their type.

### 5.4 Asymptotic Performance

For a fixed number of decoding iterations  $\ell$ , we wish to characterize the asymptotic performance as  $n \rightarrow \infty$ . The crucial observation is that the distribution on the neighborhood-shape of a randomly chosen vertex in the residual graph converges asymptotically to a multi-type branching process [23, Remark 2.13]. In our case, the multi-type branching process is defined in terms of the code parameters  $\eta$ ,  $\gamma$ , and  $\tau$ . It generalizes the (single-type) branching process described in Section 3.3 as follows. The process starts with one vertex at depth 0 which has random type  $(i, t)$  with probability  $\gamma_i \tau_t(i)$ . This vertex has neighboring (or offspring) vertices of possibly different types that extend to depth 1,

each of which has again neighboring vertices that extend to the next depth, and so on. For a vertex with type  $(i, \mathbf{t})$ , the number of offspring vertices with type  $(j, \mathbf{t}')$  is Poisson distributed with mean  $c\eta_{i,j}\gamma_j\tau_{\mathbf{t}'}(j)$ , independently of the number of offspring vertices of other types. Since the sum of independent Poisson RVs is again Poisson distributed, we have that the total number of offspring vertices of a vertex with type  $(i, \mathbf{t})$  is Poisson distributed with mean

$$\sum_{j=1}^L \sum_{\mathbf{t}'=1}^{\mathbf{t}_{\max}} c\gamma_j\eta_{i,j}\tau_{\mathbf{t}'}(j) = c \sum_{j=1}^L \gamma_j\eta_{i,j}, \quad (\text{D.38})$$

independently of  $\mathbf{t}$  (cf. (D.37)). The above multi-type branching process is denoted by  $\mathfrak{X}$ . We further use  $\mathfrak{X}(i, \mathbf{t})$  to denote the process which starts with a root vertex that has the specific type  $(i, \mathbf{t})$ .

Let  $z^{(\ell)}$  be the probability that the root vertex of the first  $\ell$  generations of  $\mathfrak{X}$  survives the  $\ell$  iterations of the peeling procedure. This probability is evaluated explicitly in Section 5.5 in terms of the code parameters  $\boldsymbol{\eta}$ ,  $\boldsymbol{\gamma}$ , and  $\boldsymbol{\tau}$ . The main result is as follows.

**Theorem 3.** *Let  $W_k$ ,  $k \in [n]$ , be the indicator RV for the event that the  $k$ -th component code of  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  declares a decoding failure after  $\ell$  iterations of decoding and define  $W = \frac{1}{n} \sum_{k=1}^n W_k$ . Then, we have*

$$\lim_{n \rightarrow \infty} \mathbb{E}[W] = z^{(\ell)}. \quad (\text{D.39})$$

Furthermore, for any  $\varepsilon > 0$ , there exist  $\delta > 0$ ,  $\beta > 0$ , and  $n_0 \in \mathbb{N}$  such that for all  $n > n_0$  we have

$$\mathbb{P}(|W - \mathbb{E}[W]| > \varepsilon) \leq e^{-\beta n^\delta}. \quad (\text{D.40})$$

*Proof.* In order to prove (D.39), we apply [23, Th. 11.6]. First, recall the following definition from [23, p. 74]. Let  $f(v, G)$  be a function defined on a pair  $(v, G)$ , where  $G$  is a graph composed of vertices with different types and  $v$  is a distinguished vertex of  $G$ , called the root. The function  $f$  is an  $\ell$ -neighborhood function if it is invariant under type-preserving rooted-graph isomorphisms and depends only on the neighborhood of the vertex  $v$  up to depth  $\ell$ . The RV  $W_k$  depends only on the depth- $\ell$  neighborhood of the  $k$ -th vertex in the residual graph of  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$ . Furthermore, the peeling outcome for the  $k$ -th vertex is invariant under isomorphisms as long as they preserve the vertex type. Hence, the RV  $W_k$  can be expressed in terms of an  $\ell$ -neighborhood function  $\mathcal{D}_\ell$  as  $W_k = \mathcal{D}_\ell(k, G)$ . That is, the function  $\mathcal{D}_\ell$  evaluates the peeling procedure on the depth- $\ell$  neighborhood of a vertex and thus determines if the corresponding component code declares a decoding failure after  $\ell$  iterations. To apply [23, Th. 11.6], we need to check that we have  $\sup_n \mathbb{E}[\mathcal{D}_\ell(k, G)^4] < \infty$ . This is true since since  $\mathcal{D}_\ell$  maps to  $\{0, 1\}$ . We then have [23, Eq. (11.4)]

$$\lim_{n \rightarrow \infty} \mathbb{E}[W] = \mathbb{E}[\mathcal{D}_\ell(\mathfrak{X})], \quad (\text{D.41})$$

where  $\mathcal{D}_\ell(\mathfrak{X})$  is defined by evaluating  $\mathcal{D}_\ell$  on the branching process  $\mathfrak{X}$  up to depth  $\ell$ , taking the initial vertex as the root. Therefore,  $\mathbb{E}[\mathcal{D}_\ell(\mathfrak{X})] = z^{(\ell)}$ . This result generalizes the convergence result (D.4) for HPCs (i.e.,  $\mathcal{G}(n, c/n)$ ) shown in Sections 4.1 and 4.2. The proof in [23] relies on a stochastic coupling of the branching process  $\mathfrak{X}$  and the neighborhood exploration process for  $\mathcal{G}^\mathcal{V}(n, \kappa)$  (which generalizes the exploration process described in Section 3.2 to handle different vertex types), see [23, Lem. 11.4] for details.

The proof of (D.40) follows along the same lines as the proof for the homogeneous case in Section 4.4, using again the typical bounded differences inequality. The bound on the maximum vertex degree for  $\mathcal{G}(n, c/n)$  in Lemma 1 applies without change also to the inhomogeneous random graph  $\mathcal{G}^\mathcal{V}(n, \kappa)$ . The only difference in the proof in Appendix A is that the equality in (D.92) becomes an inequality. The choice of the high-probability event  $\Gamma$  and the typical Lipschitz constant is then the same as described in Section 4.4.  $\square$

## 5.5 Density Evolution

In order to compute  $z^{(\ell)}$ , we proceed in a similar fashion as described in Section 4.3 and break down the computation in a recursive fashion. First, note that from the definition of  $\mathfrak{X}$  and  $\mathfrak{X}(i, t)$ , we have

$$z^{(\ell)} = \mathbb{E}[\mathcal{D}_\ell(\mathfrak{X})] = \sum_{i=1}^L \sum_{t=1}^{t_{\max}} \gamma_i \tau_t(i) z_{i,t}^{(\ell)}, \quad (\text{D.42})$$

where

$$z_{i,t}^{(\ell)} = \mathbb{E}[\mathcal{D}_\ell(\mathfrak{X}(i, t))] \quad (\text{D.43})$$

is the probability that the root vertex of the first  $\ell$  generations of the branching process  $\mathfrak{X}(i, t)$  survives the peeling procedure. We claim that

$$z_{i,t}^{(\ell)} = \Psi_{\geq t+1} \left( c \sum_{j=1}^L \sum_{t'=1}^{t_{\max}} \eta_{i,j} \gamma_j \tau_{t'}(j) x_{j,t'}^{(\ell-1)} \right), \quad (\text{D.44})$$

where  $x_{i,t}^{(\ell)}$  is recursively given by

$$x_{i,t}^{(\ell)} = \Psi_{\geq t} \left( c \sum_{j=1}^L \sum_{t'=1}^{t_{\max}} \eta_{i,j} \gamma_j \tau_{t'}(j) x_{j,t'}^{(\ell-1)} \right), \quad (\text{D.45})$$

with  $x_{i,t}^{(0)} = 1$ . The argument is the same as described in Section 4.3. In particular, to determine the survival of the root of  $\mathfrak{X}(i, t)$  after  $\ell$  peeling iterations, first determine if each offspring vertex gets removed by applying  $\ell - 1$  peeling iterations to the corresponding subtree. Then, make a decision based on the number of surviving offspring vertices. Again, one finds that offspring vertices survive independently of each other, however,

the survival probability now depends on the vertex type. In particular, in (D.44), the quantity  $x_{i,t}^{(\ell-1)}$  is the probability that a type- $(i, t)$  offspring of the root vertex survives the  $\ell - 1$  peeling iterations applied to its subtree. The argument of the function  $\Psi_{\geq t+1}$  in (D.44) is the mean number of surviving offspring vertices, which, again, is easily found to be Poisson distributed. Essentially the same arguments can be applied to find (D.45) by taking into account the connecting edge of each offspring vertex to the previous level of the tree.

Using the substitution  $x_i^{(\ell)} = \sum_{t=1}^{t_{\max}} \tau_t(i) x_{i,t}^{(\ell)}$ , it is often more convenient to express  $z^{(\ell)}$  in terms of

$$z^{(\ell)} = \sum_{i=1}^L \gamma_j \sum_{t=1}^{t_{\max}} \tau_t(i) \Psi_{\geq t+1} \left( c \sum_{j=1}^L \eta_{i,j} \gamma_j x_j^{(\ell-1)} \right), \quad (\text{D.46})$$

where

$$x_i^{(\ell)} = \sum_{t=1}^{t_{\max}} \tau_t(i) \Psi_{\geq t} \left( c \sum_{j=1}^L \eta_{i,j} \gamma_j x_j^{(\ell-1)} \right) \quad (\text{D.47})$$

with  $x_i^{(0)} = 1$  for all  $i \in [L]$ .

## 6 Discussion

Before considering a direct application of the obtained DE equations in the next section, we briefly discuss some relevant topics regarding their general application.

### 6.1 Thresholds and Code Comparisons

The decoding threshold for  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  can be defined in terms of the effective channel quality as

$$c^* \triangleq \sup\{c > 0 \mid \lim_{\ell \rightarrow \infty} z^{(\ell)} = 0\}. \quad (\text{D.48})$$

Recall that in the code construction in Section 5.1,  $\boldsymbol{\gamma}$  is assumed to be a distribution, i.e., we have  $\sum_{i=1}^L \gamma_i = 1$ . This assumption turns out to be convenient in the formulation and proof of Theorem 3 since it ensures that the number of CNs in the Tanner graph is always given by  $n$ . However, when comparing the performance of different GPCs (for example in terms of thresholds computed via (D.48)), it is more appropriate to lift this assumption and replace  $\gamma_i$  by a rescaled version  $a\gamma_i$  for all  $i$  and some constant  $a$ . This simply corresponds to scaling the total number of CNs to  $an$ .

A reasonable scaling to compare different codes is to choose  $a$  such that the effective channel quality  $c$  can be interpreted asymptotically as the average number of initial erasures in each component code, similar to HPCs in Section 2.3. Since each component

code at position  $i$  initially contains  $an_{\mathcal{C},i}c/n$  erasures, by averaging over all positions we obtain

$$\lim_{n \rightarrow \infty} a \frac{c}{n} \sum_{i=1}^L \gamma_i \left( \sum_{j \neq i} \gamma_j n \eta_{i,j} + \eta_{i,i} (\gamma_i n - 1) \right) \quad (\text{D.49})$$

$$= ac \sum_{i=1}^L \gamma_i \sum_{j=1}^L \gamma_j \eta_{i,j} = ac \boldsymbol{\gamma}^T \boldsymbol{\eta} \boldsymbol{\gamma}. \quad (\text{D.50})$$

Setting (D.50) equal to  $c$  leads to

$$a = \frac{1}{\boldsymbol{\gamma}^T \boldsymbol{\eta} \boldsymbol{\gamma}}. \quad (\text{D.51})$$

*Example 14.* For staircase codes (see Example 10), we obtain  $a = (2L - 2)/L^2$ . For large  $L$ ,  $a \approx 2/L$  so that  $a\gamma_i \approx 1/2$ .  $\triangle$

## 6.2 Upper Bound on the Decoding Threshold

An upper bound on the decoding threshold for  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$  can be given as follows, see [31, Sec. VI-A]. Assume for a moment that all component codes can correct up to  $\mathbf{t}$  erasures. The best one can hope for in this case is that each component code corrects exactly  $\mathbf{t}$  erasures. That is, in total at most  $atn$  erasures can be corrected, where  $a$  is assumed to be defined as in (D.51). Normalizing by the code length gives a maximum erasure probability of  $p \leq atn/m$  or, in terms of the effective channel quality  $c \leq atn^2/m$ . Using (D.31) as  $n \rightarrow \infty$ , we obtain  $c \leq 2\mathbf{t}$  as a necessary condition for successful decoding. This reasoning extends naturally also to the case where we allow for a mixture of erasure-correcting capabilities. In this case, one finds that  $c \leq 2\bar{\mathbf{t}}$ , where

$$\bar{\mathbf{t}} = \sum_{i=1}^L \gamma_i \sum_{\mathbf{t}=1}^{\mathbf{t}_{\max}} \tau_{\mathbf{t}}(i) \mathbf{t} \quad (\text{D.52})$$

is the mean erasure-correcting capability. This bound is used for example as a reference in the code optimization discussed in Section 7.

## 6.3 Modified Decoding Schedules

We now discuss decoding algorithms that differ from the one described in Section 2.3 and Section 5.3 in terms of scheduling. For example, for conventional PCs, one typically iterates between the component decoders for the row and column codes. Another example is the decoding of convolutional-like GPCs, such as the ones described in Examples 10 and 11. For these codes,  $L$  is typically assumed to be very large and it becomes customary to employ a sliding-window decoder. Such a decoder does not require knowledge of the entire received code array in order to start decoding. The decoder instead only operates

on a subset of the array within a so-called window configuration. After a predetermined number of iterations, this subset changes and the window “slides” to the next position.

More generally, assume that we wish to apply a different decoding schedule to  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$ . To that end, let  $\mathcal{A}^{(l)} \subseteq [L]$  for  $l \in [\ell]$  be a subset of the  $L$  CN positions. We interpret  $\mathcal{A}^{(l)}$  as *active* positions and the complementary set  $[L] \setminus \mathcal{A}^{(l)}$  as *inactive* positions in iteration  $l$ . The decoding is modified as follows. In iteration  $l$ , one only executes the BDD corresponding to CNs at active positions, i.e., positions that are contained in the set  $\mathcal{A}^{(l)}$ . CNs at inactive positions are assumed to be frozen, in the sense that they do not contribute to the decoding process. In the peeling procedure, vertices at inactive positions are simply ignored during iteration  $l$ .

In order to check if Theorem 3 remains valid for a modified decoding schedule, we adopt the convention that frozen CNs continue to declare a decoding failure if they declared a failure in the iteration in which they were last active. Moreover, we assume that each CN position belongs to the set of active positions at least once during the decoding, i.e., we assume that  $\bigcup_{l=1}^{\ell} \mathcal{A}^{(l)} = [L]$  (otherwise  $W_k$  in Theorem 3 is not defined for CNs that were never activated). Using these assumptions, it can be shown that Theorem 3 remains valid. The only difference is that the corresponding DE equations now depend on the schedule through

$$z_i^{(\ell)} = \begin{cases} \text{RHS of (D.46)} & \text{if } i \in \mathcal{A}^{(\ell)} \\ z_i^{(\ell-1)} & \text{otherwise} \end{cases}, \quad (\text{D.53})$$

and

$$x_i^{(\ell)} = \begin{cases} \text{RHS of (D.47)} & \text{if } i \in \mathcal{A}^{(\ell)} \\ x_i^{(\ell-1)} & \text{otherwise} \end{cases}. \quad (\text{D.54})$$

To see this first observe that in the proof of Theorem 3, the decoding schedule can be handled by simply assuming an appropriately modified neighborhood function  $\tilde{\mathcal{D}}_\ell$ . In particular, one may think about embedding the decoding schedule  $(\mathcal{A}^{(l)})_{l \in [\ell]}$  into the function  $\tilde{\mathcal{D}}_\ell$ . Observe that the scheduling does not change the fact that the decoding outcome is isomorphism invariant, as long as the type of all vertices is preserved. Thus, it remains to show that applying the modified decoding function  $\tilde{\mathcal{D}}_\ell$  to the branching process  $\mathfrak{X}$  results in (D.53) and (D.54). Assuming that the root vertex is active in the final iteration  $\ell$ , we can proceed as before. If, on the other hand, the root vertex is not active in the final iteration  $\ell$ , we know that the survival probability is the same as it was in the previous iteration. This gives (D.53) and applying the same reasoning for offspring vertices gives (D.54).

## 6.4 Performance on the Binary Symmetric Channel

When assuming transmission over the binary symmetric channel (BSC) as opposed to the BEC, the crucial difference is that there is a possibility that the component decoders may

miscorrect, in the sense that they introduce additional errors into the iterative decoding process. This makes a rigorous analysis challenging.

One possible approach is to change the iterative decoder. In particular, consider again the message-passing interpretation of the iterative decoding in Remark 3. In [31], the authors propose to modify the decoder in order to make the corresponding message-passing update rules extrinsic. In this case, miscorrections can be rigorously incorporated into the asymptotic decoding analysis for GPC ensembles. The reason why this approach works from a DE perspective is that for code ensembles, the entire computation graph (for a fixed depth) of a CN in the Tanner graph becomes tree-like. In fact, this makes it possible to analyze a variety of extrinsic message-passing decoders for a variety of different channels [17], including the above modified iterative decoder for the BSC.

Unfortunately, this approach appears to be limited to code ensembles. Recall that for the deterministic GPC construction  $\mathcal{C}_n(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$ , it is only the neighborhood in the residual graph that becomes tree-like (not the entire computation graph). Therefore, the independence assumption between messages is not necessarily satisfied, neither for intrinsic nor extrinsic message-passing algorithms. In general, it is not obvious how to rigorously incorporate miscorrections into an asymptotic analysis for a deterministic GPC construction. Applying our results to the BSC thus requires a similar assumption as in [9, 11, 19], i.e., either one assumes that miscorrections are negligible or that a genie prevents them.

## 7 Irregular Half-Product Codes

In this section, we consider an application of the derived DE equations for deterministic GPCs. In particular, we discuss the optimization of component code mixtures for HPCs. Recall that for (regular) HPCs, we have  $\boldsymbol{\eta} = 1$ ,  $\boldsymbol{\gamma} = 1$ , and all component codes associated with the CNs have the same erasure-correcting capability  $t$ . Similar to irregular PCs [20, 21], an *irregular* HPC is obtained by assigning component codes with different erasure-correcting capabilities to the CNs. In doing so, we hope to be able to achieve a better performance under iterative decoding compared to regular HPCs. This is of course completely analogous to other irregular code constructions, e.g., irregular LDPC codes [16].

### 7.1 Preliminaries

The assignment of erasure-correcting capabilities to the CNs is done according to the distribution  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_{t_{\max}})^\top$ . (For notational convenience, we suppress the dependence of the distribution and other quantities on the position index in the Tanner graph.) The

mean erasure-correcting capability (D.52) in this case is given by

$$\bar{\mathfrak{t}} = \sum_{\mathfrak{t}=1}^{\mathfrak{t}_{\max}} \tau_{\mathfrak{t}} \mathfrak{t}. \quad (\text{D.55})$$

The DE equation (D.47) simplifies to

$$x^{(\ell)} = \sum_{\mathfrak{t}=1}^{\mathfrak{t}_{\max}} \tau_{\mathfrak{t}} \Psi_{\geq \mathfrak{t}}(cx^{(\ell-1)}), \quad (\text{D.56})$$

with  $x^{(0)} = 1$ . The decoding threshold (D.48) can alternatively be written as

$$c^* = \sup\{c > 0 \mid \lim_{\ell \rightarrow \infty} x^{(\ell)} = 0\}, \quad (\text{D.57})$$

since  $z^{(\ell)} \rightarrow 0$  if and only if  $x^{(\ell)} \rightarrow 0$  as  $\ell \rightarrow \infty$ . From (D.56) and the fact that  $\Psi_{\geq \mathfrak{t}}(x)$  for any  $\mathfrak{t} \in \mathbb{N}$  and  $x \geq 0$  is strictly increasing, we have that the condition

$$\sum_{\mathfrak{t}=1}^{\mathfrak{t}_{\max}} \tau_{\mathfrak{t}} \Psi_{\geq \mathfrak{t}}(cx) < x, \quad \text{for } x \in (0, 1], \quad (\text{D.58})$$

implies successful decoding after a sufficiently large number of iterations, i.e., we have that  $c^* \geq c$ .

We wish to design  $\tau$  such that  $c^*$  is as large as possible. Obviously, choosing component codes with larger erasure-correcting capability gives better performance, i.e., larger thresholds. Thus, the design is done under the constraint that the mean erasure-correcting capability  $\bar{\mathfrak{t}}$  remains fixed. This is the natural analogue to the rate-constraint when designing degree distributions for irregular LDPC codes.

## 7.2 Lower Bounds on the Threshold

Before discussing the practical optimization of the distribution  $\tau$  based on a linear program in the next subsection, we show that one can construct irregular HPCs that have thresholds

$$2\bar{\mathfrak{t}} - 1 \leq c^* \leq 2\bar{\mathfrak{t}}, \quad (\text{D.59})$$

where we recall that the upper bound in (D.59) holds for any GPC according to the discussion in Section 6.2. The lower bound in (D.59) is achieved by a uniform distribution. In particular, from  $\sum_{i=1}^{\infty} \mathbb{P}(X \geq i) = \mathbb{E}[X]$ , we have

$$\sum_{\mathfrak{t}=1}^{\infty} \Psi_{\geq \mathfrak{t}}(cx) = cx, \quad (\text{D.60})$$

where we recall that  $\Psi_{\geq t}(cx) = \mathbb{P}(\text{Po}(cx) \geq t)$ . If we then choose a uniform distribution according to  $\tau_t = 1/N$  for  $t \in [N]$  (i.e.,  $t_{\max} = N$ ), we have

$$\sum_{t=1}^N \tau_t \Psi_{\geq t}(Nx) < \sum_{t=1}^{\infty} \frac{1}{N} \Psi_{\geq t}(Nx) = x \quad \text{for } x > 0, \quad (\text{D.61})$$

where the (strict) inequality follows from the fact that  $\Psi_{\geq t}(x) > 0$  for any  $t \in \mathbb{N}$  and  $x > 0$ . We see from (D.61) that the threshold for the uniform distribution satisfies  $c^* \geq N$  (cf. (D.58)). Moreover, the average erasure-correcting capability is given by

$$\bar{t} = \sum_{t=1}^N \tau_t t = \frac{1}{N} \frac{N(N+1)}{2} = \frac{N+1}{2}. \quad (\text{D.62})$$

Therefore, we have

$$2\bar{t} - c^* \leq 2 \frac{N+1}{2} - N = 1, \quad (\text{D.63})$$

or  $c^* \geq 2\bar{t} - 1$ . This simple lower bound shows that one can design irregular HPCs that are within a constant gap of the upper  $2\bar{t}$ -bound. This is in contrast to regular HPCs where  $\bar{t} = t$ . In this case, the difference between the threshold  $c^*$  and  $2t$  becomes unbounded for large  $t$ , since  $c^* = t + \sqrt{t \log t} + \mathcal{O}(\log(t))$  [34].

*Remark 14.* Essentially the same argument also allows us to give a lower bound on the threshold for irregular HPCs when the minimum erasure-correcting capability is constrained to some value  $t_{\min} > 1$ . In that case, a uniform distribution over  $\{t_{\min}, t_{\min} + 1, \dots, t_{\min} + N - 1\}$  still gives a threshold that satisfies  $c^* \geq N$ . However, we have  $\bar{t} = (N + 2t_{\min} - 1)/2$ . Hence, one obtains the lower bound  $c^* \geq 2\bar{t} - 2t_{\min} + 1$ .

### 7.3 Optimization via Linear Programming

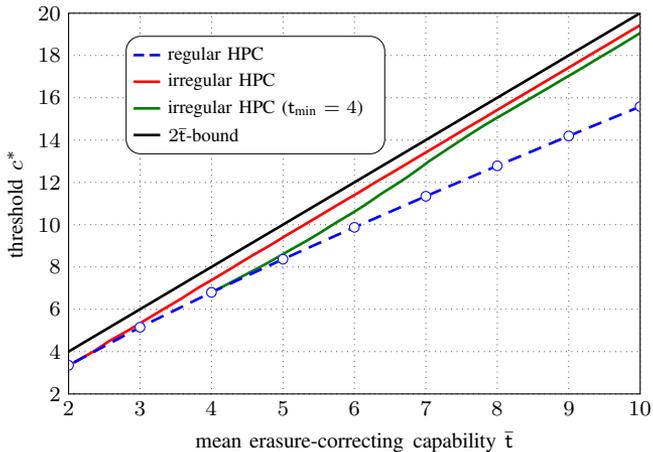
The optimal distribution maximizes the threshold  $c^*$  subject to a fixed mean erasure-correcting capability  $\bar{t}$ . Alternatively, one may fix a certain channel quality parameter  $c$  and minimize  $\bar{t}$  as follows.

$$\text{minimize}_{\tau_1, \dots, \tau_{t_{\max}}} \quad \bar{t} = \sum_{t=1}^{t_{\max}} \tau_t t \quad (\text{D.64})$$

$$\text{subject to} \quad \sum_{t=1}^{t_{\max}} \tau_t = 1, \quad \tau_1, \dots, \tau_{t_{\max}} \geq 0 \quad (\text{D.65})$$

$$\sum_{t=1}^{t_{\max}} \tau_t \Psi_{\geq t}(cx) < x, \quad x \in (0, 1]. \quad (\text{D.66})$$

The objective function and all constraints in (D.64)–(D.66) are linear in  $\tau_1, \dots, \tau_{t_{\max}}$ . Thus, after discretizing the constraint (D.66) according to  $x = i\Delta$  for  $i \in [M]$  and



**Figure 8:** Decoding thresholds for optimized irregular and regular HPCs. Thresholds for irregular HPCs are obtained via a discretized linear program with  $M = 1000$  and  $t_{\max} = 50$ .

$\Delta = 1/M$ , one obtains a linear program, which can be efficiently solved by standard numerical optimization solvers. In Fig. 8, we show the thresholds of the optimized irregular HPCs by the red line, where we used  $M = 1000$  and  $t_{\max} = 50$ , as a function of  $\bar{t}$ . We also show the thresholds for regular HPCs (where  $\bar{t} = t = 2, 3, \dots$ ) and the  $2\bar{t}$ -bound by the blue and black lines, respectively. It can be seen that the thresholds for regular HPCs diverge from the bound for large  $\bar{t}$ , as expected. Using irregular HPCs, the thresholds can be significantly improved for large  $\bar{t}$ . However, there appears to be an almost constant gap between the upper bound and the threshold curve. This gap is investigated in more detail in the next subsection.

For practical applications, it is often desirable to limit the fraction of component codes with “small” erasure-correcting capabilities in order to avoid harmful error floors [11]. It is straightforward to incorporate a minimum erasure-correcting capability  $t_{\min}$  into the above linear program. For example, the green line in Fig. 8 shows the thresholds of the optimized irregular HPCs when the minimum erasure-correcting capability is constrained to  $t_{\min} = 4$ . This additional constraint entails a threshold penalty which, however, decreases for larger values of  $\bar{t}$ .

## 7.4 Initial Component Code Loss

We now focus in more detail on the upper  $2\bar{t}$ -bound for the thresholds of irregular HPCs. In particular, we show that it is possible to give a slightly improved upper bound based on the notion of an *initial component code loss*. Based on this, it can be shown that the upper bound in (D.59) is in fact strict, i.e., for any distribution  $\tau$  with mean erasure-

correcting capability  $\bar{t}$  and threshold  $c^*$ , the gap  $2\bar{t} - c^*$  is always bounded away from zero. This gives an intuitive explanation for the gap between the threshold curve and the  $2\bar{t}$ -bound observed in Fig. 8. A similar bound for irregular LDPC code ensembles over the BEC is given in [50].

Recall that the upper bound has been derived in Section 6.2 under the (somewhat optimistic) assumption that each  $t$ -erasure correcting component code corrects exactly  $t$  erasures. In other words, each component code is assumed to contribute its maximum erasure-correcting potential to the overall decoding. A refined version of this argument takes into account the fact that a certain amount of erasure-correcting potential is lost almost surely before the iterative decoding process even begins. In particular, let the RVs  $N_{i,t}$ , for  $i = 0, 1, \dots, t-1$ , be the number of CNs corresponding to  $t$ -erasure-correcting component codes that are initially connected to  $i < t$  erased VNs. In the first decoding iteration, each of these CNs corrects only  $i$  erasures instead of  $t$ , i.e., the maximum number of erasures  $E$  that we can hope to correct is upper bounded by

$$E \leq \bar{t}n - \sum_{t=1}^{t_{\max}} \sum_{i=0}^{t-1} N_{i,t}(t-i). \quad (\text{D.67})$$

Since  $E/n$  and  $N_{i,t}/n$  converge almost surely to the deterministic values  $c/2$  and  $\tau_t \Psi_{=i}(c)$ , respectively, we obtain

$$c \leq 2\bar{t} - 2\mathcal{L}_{\tau}(c) \quad (\text{D.68})$$

as a necessary condition for successful decoding, where we implicitly defined the initial component code loss for the distribution  $\tau$  as

$$\mathcal{L}_{\tau}(c) \triangleq \sum_{t=1}^{t_{\max}} \tau_t \mathcal{L}(t, c) \quad (\text{D.69})$$

with

$$\mathcal{L}(t, c) \triangleq \sum_{i=0}^{t-1} \Psi_{=i}(c)(t-i) \quad (\text{D.70})$$

for  $c > 0$  and  $t \in \mathbb{N}$ .

*Remark 15.* The affine extension of  $\mathcal{L}(t, c)$  for a fixed  $c \geq 0$  is convex in  $t \in [1; \infty)$  in the sense that for any  $c \geq 0$  and  $t = 2, 3, \dots$ , we have

$$\mathcal{L}(t-1, c) + \mathcal{L}(t+1, c) = 2\mathcal{L}(t, c) + \Psi_{=t}(c) \quad (\text{D.71})$$

$$\geq 2\mathcal{L}(t, c). \quad (\text{D.72})$$

This implies that for any distribution  $\tau$  with average erasure-correcting capability  $\bar{t}$ , the associated initial component code loss satisfies

$$\mathcal{L}_{\tau}(c) \geq \mathcal{L}(\lfloor \bar{t} \rfloor, c), \quad (\text{D.73})$$

i.e., the initial loss is minimized for regular HPCs.

The bound (D.68) has a natural interpretation in terms of areas related to the curves involved in the condition (D.58), similar to the area theorem for irregular LDPC code ensembles. Indeed, an alternative way to show that successful decoding implies (D.68) is by integrating the condition (D.58). Using integration by parts, one obtains the indefinite integral [51]

$$\int \Psi_{\geq t}(x) dx = x\Psi_{\geq t}(x) + t\Psi_{\leq t}(x). \quad (\text{D.74})$$

Thus, we have

$$c \int_0^1 \Psi_{\geq t}(cx) dx = c\Psi_{\geq t}(c) + t\Psi_{\leq t}(c) - t \quad (\text{D.75})$$

$$= c(1 - \Psi_{< t}(c)) + t\Psi_{\leq t}(c) - t \quad (\text{D.76})$$

$$= c - t + \mathcal{L}(t, c), \quad (\text{D.77})$$

where the last equality follows from

$$t\Psi_{\leq t}(c) - c\Psi_{< t}(c) = t\Psi_{\leq t}(c) - c \sum_{k=0}^{t-1} \frac{c^k}{k!} e^{-c} \quad (\text{D.78})$$

$$= t\Psi_{\leq t}(c) - c \sum_{k=1}^t \frac{c^{k-1}}{(k-1)!} e^{-c} \quad (\text{D.79})$$

$$= t\Psi_{\leq t}(c) - \sum_{k=0}^t \frac{c^k}{k!} e^{-c} k \quad (\text{D.80})$$

$$= \sum_{k=0}^t \Psi_{=k}(c)(t - k) \quad (\text{D.81})$$

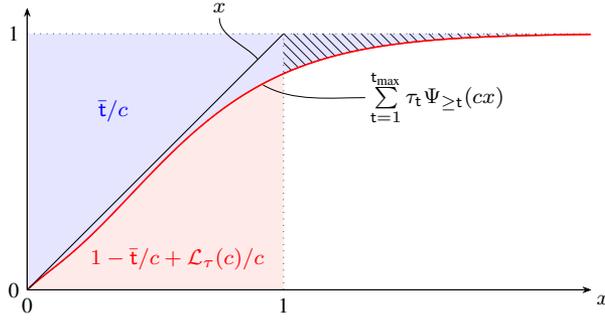
$$= \mathcal{L}(t, c). \quad (\text{D.82})$$

Hence, integrating both sides of (D.58) from zero to one and using (D.77), one obtains

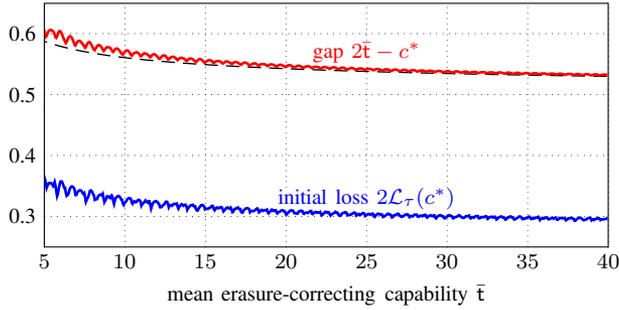
$$\frac{1}{c} \sum_{t=1}^{t_{\max}} \tau_t (c - t + \mathcal{L}(t, c)) < \frac{1}{2}, \quad (\text{D.83})$$

or, equivalently, (D.68).

A visualization is shown in Fig. 9, where the red and black lines correspond to the left-hand side (LHS) and RHS of (D.58), respectively. The area below the red curve up to  $x = 1$  (shown in red) corresponds to the LHS of (D.83). Similarly, it can be shown using (D.74) that the area between the red line and  $x = 1$  (shown in blue) corresponds to the scaled erasure-correcting capability  $\bar{t}/c$ . Note that the  $2\bar{t}$ -bound on the threshold simply corresponds to the fact that the blue area cannot be smaller than  $1/2$ , since otherwise the red and black lines would have to cross. From the previous discussion, we have seen



**Figure 9:** Graphical interpretation of the upper threshold bounds.



**Figure 10:** Gap between the threshold  $c^*$  and the  $2\bar{t}$ -bound for the optimized irregular HPCs.

that the gap to the upper  $2\bar{t}$ -bound is partially due to the initial component code loss. In particular, by combining the blue and red areas, it can be seen that the hatched area in Fig. 9 corresponds precisely to the (scaled) loss  $\mathcal{L}_\tau(c)/c$ .

Consider now again the outcome of the linear program for the optimized irregular HPCs in Fig. 8. In Fig. 10, the (vertical) gap  $2\bar{t} - c^*$  between the black and red lines in Fig. 8 is shown for a larger range of  $\bar{t}$ . It can be seen that the gap is decreasing with  $\bar{t}$ , albeit rather slowly. We also plot the initial component code loss for the optimized distributions at the threshold value by the blue line. From this, we see that the initial component code loss accounts for approximately half of the threshold gap for the optimized irregular distributions.

*Remark 16.* In fact, we conjecture that the following is true. Assume  $c \in \mathbb{N}$ . Then, for any distribution  $\tau$  with threshold  $c^* \geq c$  and mean erasure-correcting capability  $\bar{t}$ , we have

$$\bar{t} \geq \frac{c}{2} + \frac{1}{c} \sum_{t=1}^c \mathcal{L}(t, c). \quad (\text{D.84})$$

This bound is shown in Fig. 10 by the dashed line, although we failed to prove it. Proving (D.84) would be interesting, since one can show that  $\lim_{c \rightarrow \infty} \frac{1}{c} \sum_{t=1}^c \mathcal{L}(t, c) = 1/4$  and hence  $2\bar{t} - c^* \geq 1/2$ , which seems to be the constant to which the optimization outcome is converging for  $\bar{t} \rightarrow \infty$ .

## 7.5 Simulation Results

In order to illustrate how the thresholds can be used to design practical irregular HPCs, we consider (shortened) binary BCH codes as component codes. Given the Galois-field extension degree  $\nu$ , a shortening parameter  $s$ , and the erasure-correcting capability  $t$ , we let the component code be an  $(n, k_C, d_{\min})$  BCH code, where  $n = 2^\nu - 1 - s$ ,  $d_{\min} = t + 1$ , and

$$k_C = \begin{cases} n - \nu t/2, & t \text{ even} \\ n - \nu(t-1)/2 - 1, & t \text{ odd} \end{cases}. \quad (\text{D.85})$$

In the following, we consider two irregular HPCs, where  $\bar{t} \approx 7$ . As a comparison, we use a regular HPC with  $\tau_7 = 1$  for which  $c^* \approx 11.34$ . The optimal distribution (rounded to three decimal places) according to the linear program (D.64)–(D.66) is given by

$$\begin{aligned} \tau_1 &= 0.070, & \tau_2 &= 0.103, & \tau_4 &= 0.115, \\ \tau_5 &= 0.179, & \tau_{10} &= 0.496, & \tau_{11} &= 0.037, \end{aligned} \quad (\text{D.86})$$

which yields  $c^* \approx 13.42$ . We also consider the case where the minimum erasure-correcting capability is constrained to be  $t_{\min} = 4$ . For this case, one obtains

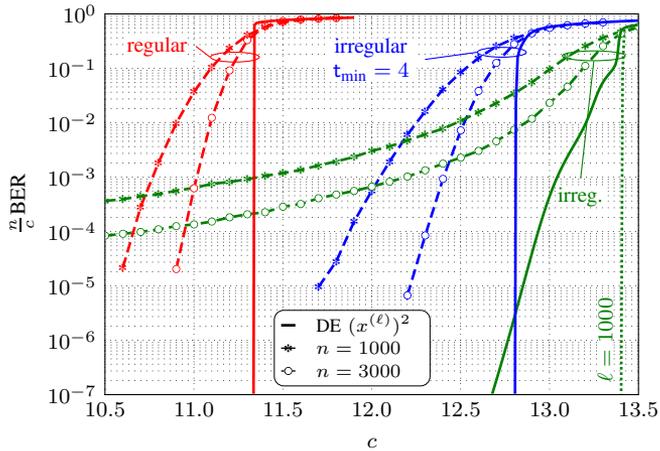
$$\tau_4 = 0.495, \quad \tau_9 = 0.029, \quad \tau_{10} = 0.476, \quad (\text{D.87})$$

and the threshold is reduced to  $c^* \approx 12.88$ .

For the simulations, we consider two different component code lengths,  $n = 1000$  (i.e.,  $\nu = 10$  and  $s = 23$ ) and  $n = 3000$  (i.e.,  $\nu = 12$  and  $s = 1095$ ), leading to an overall length of the HPCs of  $m \approx 500,000$  and  $m \approx 4,500,000$ , respectively. If we denote the dimension of the  $k$ -th component code by  $k_{C_k}$ , the code rate is lower bounded by [7, Sec. 5.2.1]

$$R \geq 1 - \frac{\sum_{k=1}^n (n - k_{C_k})}{m}. \quad (\text{D.88})$$

For the regular case and the distributions (D.86) and (D.87), the lower bound evaluates to approximately 0.93 and 0.97 for  $n = 1000$  and  $n = 3000$ , respectively. (In order to obtain shorter (longer) codes for the same rate, one needs to reduce (increase)  $\bar{t}$ .) Although the chosen values for  $n$  are merely for illustration purposes, we remark that the delay caused by such seemingly long block-lengths is typically not a problem for high-speed applications. For example, the delay for the GPCs designed for fiber-optical communication systems in [8, 13] is in the order of 2,000,000 bits.



**Figure 11:** Simulation results (dashed lines) for regular and optimized irregular HPCs for two different values of  $n$  and  $\ell = 100$ . DE results (solid lines) are shown for  $\ell = 100$ .

Simulation results are shown in Fig. 11 by the dashed lines. In all cases, the maximum number of decoding iterations is restricted to  $\ell = 100$ . Results for regular HPCs are shown in red, while results for the irregular HPCs defined by the optimized distributions (D.86) and (D.87) are shown in green and blue, respectively. For lower error rates, the irregular HPCs defined by (D.86) are clearly outperformed by regular HPCs and HPCs defined by the distribution (D.87). This is due to the relatively large fraction of component codes that only correct 1 and 2 erasures, which leads to a large error floor.

It is interesting to inspect the DE predictions for  $\ell = 100$ , which are shown by the solid lines in Fig. 11. The predicted performance for the regular and irregular distribution (D.87) drops sharply, while the predicted performance for the distribution (D.86) shows a markedly different behavior due to the finite iteration number. It is therefore important to stress that an optimization via the condition (D.58) implicitly assumes an unrestricted number of decoding iterations. (As a reference, the DE prediction for the distribution (D.86) with  $\ell = 1000$  is shown by the green dotted line.) Thus, if we had done an optimization based on DE assuming  $\ell = 100$  and targeting an error rate of around  $10^{-7}$  in Fig. 11, we would have rejected the distribution (D.86) in favor of the distribution (D.87) right away. However, it is not obvious if such an optimization admits a formulation in terms of a linear program, and therefore it would very likely have to be based on a heuristic optimization procedure.

Lastly, the HPCs defined by (D.87) have a comparable finite-length scaling behavior below the threshold and no noticeable error-floor for the simulated error rates. As a consequence, the performance gains for this distribution over the regular HPCs predicted by DE are well preserved also for finite lengths.

## 8 Conclusions and Future Work

In this paper, we studied the performance of deterministically constructed GPCs under iterative decoding. Using the framework of sparse inhomogeneous random graphs, we showed how to derive the DE equations that govern the asymptotic behavior. In principle, DE can be used for a variety of different applications, e.g., parameter tuning, optimization of decoding schedules, or the design of new GPCs. Here, we used the derived DE equations to optimize irregular HPCs that employ a mixture of component codes with different erasure-correcting capabilities. Using an approach based on linear programming, we obtained irregular HPCs that outperform regular HPCs.

For future work, it would be interesting to analyze deterministic code constructions that incorporate VNs with larger degrees. Larger VN degrees are easily incorporated into an ensemble approach, see, e.g., [33]. An example of a corresponding deterministic code construction is the case where code arrays are generalized from two to three (or higher) dimensional objects, e.g., a cube-shaped code array. In that case, the residual graph could be modeled as a random hypergraph. Cores in random hypergraphs have for example been studied in [52].

## A Proof of Lemma 1

First, we upper-bound the probability that the degree  $D_k$  of the  $k$ -th vertex exceeds  $d_n$  using the Chernoff bound. Let  $\theta \sim \mathbf{B}(c/n)$ , then, for any  $\lambda > 0$ , we have

$$\mathbb{P}(D_k \geq d_n) = \mathbb{P}(e^{\lambda D_k} \geq e^{\lambda d_n}) \tag{D.89}$$

$$\leq e^{-\lambda d_n} \mathbb{E}[e^{\lambda D_k}] \tag{D.90}$$

$$= e^{-\lambda d_n} \mathbb{E}\left[e^{\lambda(\theta_{k,1} + \dots + \theta_{k,n})}\right] \tag{D.91}$$

$$= e^{-\lambda d_n} (\mathbb{E}[e^{\lambda \theta}])^{n-1} \tag{D.92}$$

$$= e^{-\lambda d_n} (1 - p + pe^\lambda)^{n-1} \tag{D.93}$$

$$\leq e^{-\lambda d_n} \left(1 + \frac{c}{n} (e^\lambda - 1)\right)^n \tag{D.94}$$

$$\leq e^{-\lambda d_n} e^{c(e^\lambda - 1)} \tag{D.95}$$

$$\leq e^{-c - d_n \ln \frac{d_n}{ce}} \tag{D.96}$$

where (D.90) follows from applying Markov's inequality, (D.92) holds because all  $\theta_{k,j} \sim \theta$  are independent except  $\theta_{k,k} = 0$ , (D.95) stems from  $(1+x/n)^n \leq e^x$  for  $x \geq 0$ , and (D.96) follows from minimizing over  $\lambda$ . Thus, for  $d_n = \Omega(\log(n))$  and any  $\beta > 0$ , there is an  $n_0$  such that  $\mathbb{P}(D_k \geq d_n) \leq e^{-\beta d_n}$ . Hence, if one chooses  $\beta$  large enough, then the union

bound implies

$$\mathbb{P}(D_{\max} \geq d_n) \leq n\mathbb{P}(D_k \geq d_n) \tag{D.97}$$

$$\leq ne^{-\beta d_n} \tag{D.98}$$

$$= e^{\log(n) - \beta d_n} \tag{D.99}$$

$$= e^{-\beta(d_n - \log(n)/\beta)} \tag{D.100}$$

$$\leq e^{-\beta d_n/2} \tag{D.101}$$

for all  $n \geq n_0$ .

## B Bound on the Second Moment of $T_\ell$

To obtain a bound on  $\mathbb{E}[T_\ell^2]$ , we first show how to compute the corresponding quantity  $\mathbb{E}[\bar{T}_\ell^2]$  for a branching process. The final expression is a function of the mean  $\mu_{\bar{\xi}}$  and variance  $\sigma_{\bar{\xi}}^2$  of the offspring distribution  $\bar{\xi}$ .

First, from  $\bar{T}_\ell = \bar{Z}_0 + \bar{Z}_1 + \dots + \bar{Z}_\ell$ , we obtain

$$\mathbb{E}[\bar{T}_\ell^2] = \sum_{i=0}^{\ell} \mathbb{E}[\bar{Z}_i^2] + 2 \sum_{i=1}^{\ell} \sum_{j=0}^{i-1} \mathbb{E}[\bar{Z}_i \bar{Z}_j]. \tag{D.102}$$

Using the definition of  $\bar{Z}_i$  and the law of total expectation, it can be shown that for  $i > j$ , we have

$$\mathbb{E}[\bar{Z}_i \bar{Z}_j] = \mu_{\bar{\xi}}^{i-j} \mathbb{E}[\bar{Z}_j^2]. \tag{D.103}$$

Inserting (D.103) into (D.102) leads to

$$\mathbb{E}[\bar{T}_\ell^2] = \sum_{i=0}^{\ell} \mathbb{E}[\bar{Z}_i^2] + 2 \sum_{i=1}^{\ell} \sum_{j=0}^{i-1} \mu_{\bar{\xi}}^{i-j} \mathbb{E}[\bar{Z}_j^2]. \tag{D.104}$$

Next, we can use the well-known expressions for the mean and variance of  $\bar{Z}_i$  (see, e.g., [42, p. 396]) to obtain

$$\mathbb{E}[\bar{Z}_\ell^2] = \begin{cases} \sigma_{\bar{\xi}}^2 \mu_{\bar{\xi}}^{\ell-1} \frac{\mu_{\bar{\xi}}^{\ell}-1}{\mu_{\bar{\xi}}-1} + \mu_{\bar{\xi}}^{2\ell}, & \mu_{\bar{\xi}} \neq 1 \\ \ell \sigma_{\bar{\xi}}^2 + 1, & \mu_{\bar{\xi}} = 1 \end{cases}. \tag{D.105}$$

Inserting (D.105) into (D.104) leads to the desired explicit characterization of  $\mathbb{E}[\bar{T}_\ell^2]$ . Of particular interest here is the case where the offspring distribution is Poisson with mean  $c$ . In this case, we have  $\mu_{\bar{\xi}} = c$  and  $\sigma_{\bar{\xi}}^2 = c$ , which leads to

$$\mathbb{E}[\bar{T}_\ell^2] = \begin{cases} \frac{c^{2\ell+3} - 1 - (2\ell+3)c^\ell(c-1)}{(c-1)^3}, & c \neq 1 \\ \frac{(\ell+1)(\ell+2)(2\ell+3)}{6}, & c = 1 \end{cases}. \tag{D.106}$$

Finally, using the same steps as in the proof of [41, Th. 4.2] and [41, Th. 3.20] one can show that (D.106) is an upper bound on  $\mathbb{E}[T_\ell^2]$ , i.e., we have  $\mathbb{E}[T_\ell^2] \leq \mathbb{E}[\bar{T}_\ell^2]$ .

## References

- [1] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [2] G. D. Forney, Jr., “Concatenated codes,” Ph.D. dissertation, Massachusetts Institute of Technology, 1965.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Proc. IEEE Int. Conf. Communications (ICC)*, Geneva, Switzerland, 1993.
- [4] P. Elias, “Error-free coding,” *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Apr. 1954.
- [5] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
- [6] N. Abramson, “Cascade decoding of cyclic product codes,” *IEEE Trans. Commun. Tech.*, vol. 16, no. 3, pp. 398–402, Jun. 1968.
- [7] W. Ryan and S. Lin, *Channel Codes Classical and Modern*. Cambridge University, 2009.
- [8] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase codes: FEC for 100 Gb/s OTN,” *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.
- [9] J. Justesen and T. Høholdt, “Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes,” in *Proc. IEEE Information Theory Workshop (ITW)*, Tahoe City, CA, 2007.
- [10] J. Justesen, K. J. Larsen, and L. A. Pedersen, “Error correcting coding for OTN,” *IEEE Commun. Mag.*, vol. 59, no. 9, pp. 70–75, Sep. 2010.
- [11] J. Justesen, “Performance of product codes and related structures with iterated decoding,” *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.
- [12] M. Scholten, T. Coe, and J. Dillard, “Continuously-interleaved BCH (CI-BCH) FEC delivers best in class NECG for 40G and 100G metro applications,” in *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, CA, 2010.

- [13] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Matahreh, “Iterative hard-decision decoding of braided BCH codes for high-speed optical communication,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.
- [14] L. M. Zhang and F. R. Kschischang, “Staircase codes with 6% to 33% overhead,” *J. Lightw. Technol.*, vol. 32, no. 10, pp. 1999–2002, May 2014.
- [15] C. Häger, A. Graell i Amat, H. D. Pfister, A. Alvarado, F. Brännström, and E. Agrell, “On parameter optimization for staircase codes,” in *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA, 2015.
- [16] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient erasure correcting codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.
- [17] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [18] R. Gallager, “Low-density parity-check codes,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1963.
- [19] M. Schwartz, P. Siegel, and A. Vardy, “On the asymptotic performance of iterative decoders for product codes,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Adelaide, SA, 2005.
- [20] S. Hirasawa, M. Kasahara, Y. Sugiyama, and T. Namekawa, “Modified product codes,” *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 299–306, Mar. 1984.
- [21] M. Alipour, O. Etesami, G. Maatouk, and A. Shokrollahi, “Irregular product codes,” in *Proc. IEEE Information Theory Workshop (ITW)*, Lausanne, Switzerland, 2012.
- [22] A. J. Feltström, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, “Braided block codes,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jul. 2009.
- [23] B. Bollobás, S. Janson, and O. Riordan, “The phase transition in inhomogeneous random graphs,” *Random Structures and Algorithms*, vol. 31, no. 1, pp. 3–122, Aug. 2007.
- [24] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.
- [25] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, “Analysis of random processes via and-or tree evaluation,” in *Proc. 9th Annual ACM-SIAM Symp. Discrete Algorithms*, San Francisco, CA, 1998.

- [26] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, “Iterative decoding threshold analysis for LDPC convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.
- [27] M. Lentmaier, M. B. Tavares, and G. P. Fettweis, “Exact erasure channel density evolution for protograph-based generalized LDPC codes,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seoul, South Korea, Jun. 2009.
- [28] M. Lentmaier and G. P. Fettweis, “On the thresholds of generalized LDPC convolutional codes based on protographs,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Austin, TX, 2010.
- [29] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN Progress Report 42-154*, JPL, 2005.
- [30] N. Miladinovic and M. Fossorier, “Generalized LDPC codes and generalized stopping sets,” *IEEE Trans. Commun.*, vol. 56, no. 2, pp. 201–212, Feb. 2008.
- [31] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high rates with iterative hard-decision decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [32] —, “Approaching capacity at high-rates with iterative hard-decision decoding,” *arxiv:1202.6095v3 [cs.IT]*, May 2015. [Online]. Available: <http://arxiv.org/abs/1202.6095.pdf>
- [33] L. M. Zhang, D. Truhachev, and F. R. Kschischang, “Spatially-coupled split-component codes with bounded-distance component decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Hong Kong, 2015.
- [34] B. Pittel, J. Spencer, and N. Wormald, “Sudden emergence of a giant  $k$ -core in a random graph,” *Journal of Combinatorial Theory, Series B*, vol. 67, no. 1, pp. 111–151, May 1996.
- [35] P. Erdős and A. Rényi, “On random graphs I,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
- [36] E. N. Gilbert, “Random graphs,” *Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, Dec. 1959.
- [37] H. D. Pfister, S. K. Emmadi, and K. Narayanan, “Symmetric product codes,” in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, 2015.
- [38] B. Bollobás and Riordan O., “Random graphs and branching processes,” in *Handbook of Large-Scale Random Networks*, 2009, pp. 15–115.

- [39] O. Riordan, “The  $k$ -core and branching processes,” *Combinatorics, Probability and Computing*, vol. 17, no. 1, pp. 111–136, Jun. 2007.
- [40] N. Alon and J. H. Spencer, *The Probabilistic Method*, 2nd ed. Wiley-Interscience, Aug. 2000.
- [41] R. van der Hofstad, “Random graphs and complex networks. Vol. I,” 2014.
- [42] S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes*, 2nd ed. Academic Press, 1975.
- [43] A. Dembo and A. Montanari, “Ising models on locally tree-like graphs,” *Ann. Appl. Probab.*, vol. 20, no. 2, pp. 565–592, 2010.
- [44] A. Montanari, “Statistical mechanics and algorithms on sparse and random graphs,” Oct. 2014.
- [45] M. G. Luby, M. Mitzenmacher, M. A. Shokrollah, and D. A. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” in *Proc. AMC Symp. on Theory of Computing (STOC)*, New York, USA, 1998.
- [46] J. S. Rosenthal, *A First Look At Rigorous Probability Theory*, 2nd ed. World Scientific, 2006.
- [47] L. Warnke, “On the method of typical bounded differences,” *Combinatorics, Probability and Computing (to appear)*, 2015. [Online]. Available: <http://arxiv.org/pdf/1212.5796.pdf>
- [48] B. P. Smith, “Error-correcting codes for fibre-optic communication systems,” Ph.D. dissertation, University of Toronto, 2011.
- [49] B. Söderberg, “General formalism for inhomogeneous random graphs,” *Phys. Rev. E*, vol. 66, no. 6, Dec. 2002.
- [50] M. A. Shokrollahi, “New sequences of linear time erasure codes approaching the channel capacity,” in *Proc. Int. Symp. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 1999.
- [51] I. Gradshteyn and D. Ryzhik, *Table of Integrals, Series, and Products, Seventh Edition*, 7th ed. Academic Press, 2007.
- [52] M. Molloy, “Cores in random hypergraphs and boolean formulas,” *Random Structures and Algorithms*, vol. 27, no. 1, pp. 124–135, Aug. 2005.



**Density Evolution and Error Floor Analysis for Staircase and Braided Codes**

Christian Häger, Henry D. Pfister, Alexandre Graell i Amat, and Fredrik Brännström

in *Proc. Optical Fiber Communication Conf. and Exposition (OFC)*,  
Anaheim, CA, 2016

*The layout has been revised.*

## Abstract

We analyze deterministically constructed (i.e., non-ensemble-based) codes in the waterfall and error floor region. The analysis directly applies to several FEC classes proposed for high-speed OTNs such as staircase and braided codes.

## 1 Introduction

Several authors have proposed improvements over the forward error correction (FEC) codes for optical transport networks (OTNs) from the ITU-T G.975 and G.975.1 recommendations. Some of these proposals, e.g., staircase codes (SCs) [1] and braided codes (BCs) [2], are extensions of classical product codes (PCs) and we refer to them as generalized PCs (GPCs). GPCs are particularly suited for high-speed applications due to their lower complexity under iterative hard-decision decoding compared to message-passing decoding of low-density parity-check (LDPC) codes [1].

FEC design requires assessment of nontrivial trade-offs between performance, complexity, and decoding delay. Identifying these trade-offs is greatly simplified with the availability of theoretical tools that allow the prediction of the post-FEC bit error rate (BER) performance without resorting to time-consuming Monte-Carlo simulations. In this paper, we propose a deterministic GPC construction which encompasses PCs, staircase codes (SCs), and (block-wise) BCs as special cases. The main contribution is a characterization of the performance under iterative decoding in the waterfall region by means of a density evolution (DE) analysis. Our work generalizes previous work in [3, 4] to a large class of GPCs. Even though other classes of GPCs are also discussed in [4], the DE analysis in these papers is limited to PCs and their symmetric subcodes (so-called half-PCs).

As an application, we present a case study comparing SCs and two variants of BCs. Supplemented with an error floor analysis, we show that the symmetric subcode of a BC can outperform both SCs and conventional BCs in the waterfall region, at a lower error floor and decoding delay.

## 2 Density Evolution for Deterministic GPCs

We denote a GPC by  $\mathcal{C}_n(\eta)$ , where  $n$  is the number of constraint nodes (CNs) in the underlying Tanner graph and  $\eta$  is a binary, symmetric  $L \times L$  matrix that defines the graph connectivity. Since GPCs have a natural representation in terms of two-dimensional code arrays (see, e.g., Fig. 1), one may alternatively think about  $\eta$  as specifying the array shape. We will see in the following that different choices for  $\eta$  recover well-known code classes.

## 2.1 Code Construction

To construct the Tanner graph that defines  $\mathcal{C}_n(\eta)$ , assume that there are  $L$  positions. Then, place  $d \triangleq n/L$  CNs at each position and connect each CN at position  $i$  to each CN at position  $j$  through a variable node (VN) if and only if  $\eta_{i,j} = 1$ .

*Example:* A PC is obtained for  $L = 2$  and  $\eta = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ . The two positions correspond to “row” and “column” codes, respectively. The code array is of size  $d \times d$ , where  $d = n/2$ . Each of the  $d^2$  VNs corresponds to one bit in the array.  $\triangle$

CNs at position  $i$  have degree  $d \sum_{j \neq i} \eta_{i,j} + \eta_{i,i}(d - 1)$ , where the second term arises from the fact that we cannot connect a CN to itself if  $\eta_{i,i} = 1$ . Recall that the CN degree specifies the length of the component code associated with the CN. Here we assume that each CN corresponds to a  $t$ -error correcting Bose–Chaudhuri–Hocquenghem (BCH) code.

## 2.2 Iterative Decoding

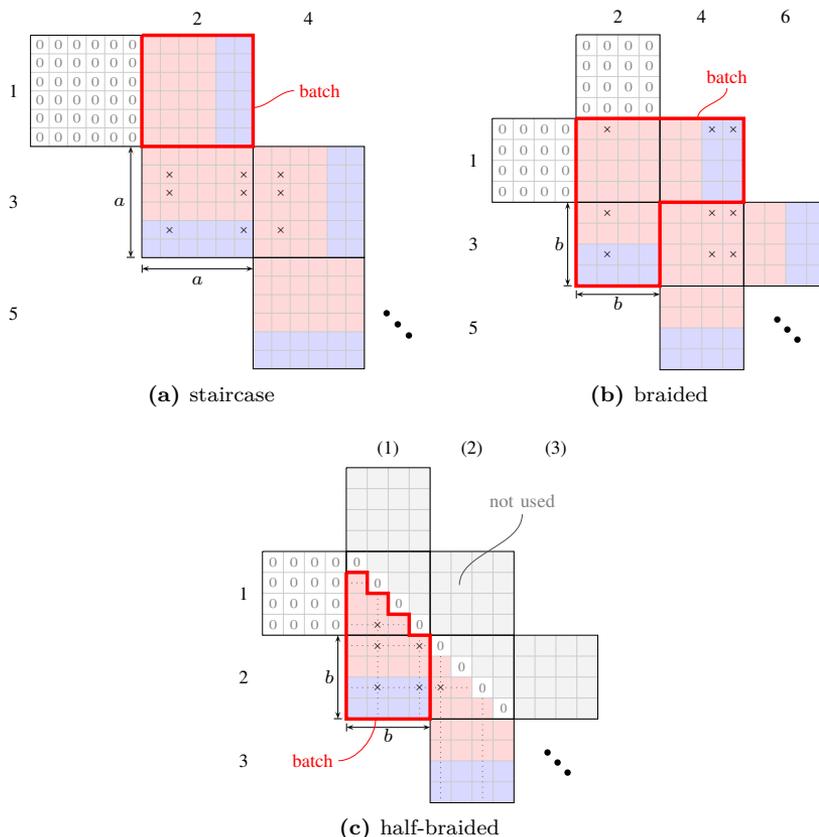
Suppose that a codeword of  $\mathcal{C}_n(\eta)$  is transmitted over a binary symmetric channel with crossover probability  $p$ . The decoding is performed iteratively assuming  $\ell$  iterations according to the following procedure. In each iteration, perform bounded-distance decoding (BDD) of all component codes and update the bits of the associated VNs according to the decoding outcome. For simplicity, we consider idealized BDD similar to [3, 4], which works as follows. If the Hamming weight of the error pattern is less or equal to  $t$ , the pattern is corrected. If the weight exceeds  $t$ , the component code declares an error but leaves all associated bits unchanged.

## 2.3 Density Evolution

We wish to characterize the asymptotic decoding performance in the limit  $n \rightarrow \infty$ . To that end, let  $\Psi_{\geq t}(\lambda) \triangleq 1 - e^{-\lambda} \sum_{i=0}^{t-1} \frac{\lambda^i}{i!}$  be the tail probability of a Poisson random variable,  $p = c/n$  for some  $c > 0$ , and assume that we compute

$$z_i^{(\ell)} = \Psi_{\geq t+1} \left( \frac{c}{L} \sum_{j=1}^L \eta_{i,j} x_j^{(\ell-1)} \right), \quad \text{where} \quad x_i^{(\ell)} = \Psi_{\geq t} \left( \frac{c}{L} \sum_{j=1}^L \eta_{i,j} x_j^{(\ell-1)} \right) \quad (\text{E.1})$$

for  $i \in \{1, \dots, L\}$  with  $x_i^{(0)} = 1$ . The main technical result is as follows. Let the random variable  $Z$  be the fraction of component codes that declare errors in iteration  $\ell$ . Then,  $Z$  converges almost surely to  $\frac{1}{L} \sum_{i=1}^L z_i^{(\ell)}$  as  $n \rightarrow \infty$ . In other words, the code performance concentrates around a deterministic value computed by the recursion in (E.1) for sufficiently large  $n$ . This result is analogous to the DE analysis for LDPC codes and for a formal proof we refer the reader to [5]. Observe that this result applies to the deterministically constructed code  $\mathcal{C}_n(\eta)$  and does not rely on the definition of a code ensemble (i.e., a set of codes), which is typically required to apply DE. We remark that the analysis can be modified to account for different decoding schedules, e.g., alternations between “rows” and “columns” or window decoders (or a combination thereof), see [5].



**Figure 1:** Code arrays using the same component code with length  $n_c = 12$  and dimension  $k_c = 10$ .

The above result is asymptotic in  $n$ . In practice,  $n$  is typically “large enough” and  $d = n/L$  exceeds several hundreds of bits. We can then use (E.1) to predict the region where the post-FEC BER curve of  $\mathcal{C}_n(\eta)$  bends into the characteristic waterfall behavior by using  $\text{BER} \approx p\mathbf{x}\eta\mathbf{x}^T/\|\eta\|_F^2$ , where  $\|\eta\|_F^2$  is the number of 1s in  $\eta$  and  $\mathbf{x} \triangleq (x_1^{(\ell)}, \dots, x_L^{(\ell)})$ .

### 3 Case Study: Comparison of Staircase, Braided, and Half-Braided Codes

Consider the code arrays in Fig. 1, where (a) corresponds to a SC [1], (b) to a variant of a block-wise BC [6], and (c) to a half-BC [4, 7]. The definition of these codes is usually easiest to understand in terms of their systematic encoding procedure. For the

SC and BC, the red array elements are filled with information bits and the blue array elements are the parity bits obtained by systematically encoding rows and/or columns. The procedure is essentially the same for the half-BC, however, each component code acts on an L-shape, i.e., both a partial row and column, which includes the zero on the array diagonal. Equivalently, one can encode the BC with a zero diagonal and symmetrically placed information bits with respect to the diagonal. Thus, the half-BC code can be viewed as a (punctured) subcode of the BC.

By considering the Tanner graph of all three codes, one may check that they can be seen as special cases of  $\mathcal{C}_n(\eta)$ . In particular, the SC is recovered for  $\eta_{i,i+1} = \eta_{i+1,i} = 1$  for  $i \in \{1, \dots, L-1\}$  and zeros elsewhere. The BC has the same  $\eta$ , but additionally  $\eta_{2i-1,2i+2} = \eta_{2i+2,2i-1} = 1$  for  $i \in \{1, \dots, L/2-1\}$ . The half-BC has the same  $\eta$  as the SC, but additionally  $\eta_{i,i} = 1$  for  $i \in \{1, \dots, L\}$ . The numbers in Fig. 1 indicate the positions in the construction in Sec. 2.1.

### 3.1 Parameters and Error Floor

We use a BCH component code with parameters  $(n_c, k_c, t) = (720, 690, 3)$ , where  $n_c$  and  $k_c$  are the length and dimension of the code. All other parameters are listed in Table 1 and briefly discussed in the following. Encoding is performed in *batches*, where one batch is indicated by the thick, red lines in Fig. 1, and  $B$  denotes the number of bits per batch. The code rate  $R$  is the ratio between the number of information bits per batch and  $B$ . All three codes have roughly the same rate  $R \approx 0.917$  (and an FEC overhead of 9.1%). The decoding is performed in a sliding-window fashion, where each window comprises  $W$  received batches. The decoding delay (in bits) is given by  $D = WB$  and we have chosen  $W$  such that the SC and BC have the same delay, which is roughly twice as much as for the half-BC. The SC and BC are decoded by iterating  $\ell$  times between rows and columns within each window. For the half-BC, all component codes within each window are decoded simultaneously and  $\ell$  is increased to keep the same decoding complexity (note that the number of component codes per window is reduced by half for half-BCs).

To estimate the error floor, we follow the analysis and terminology in [1, Sec. V-B]. Let  $s_{\min}$  be the size of the minimal stall pattern, defined as the minimum number of array positions which, when all received in error, cause the decoder to stall. Examples are shown by the crosses in Fig. 1 for  $t = 2$ . A stall pattern is said to be assigned to a batch if at least one of its array positions belongs to the batch and no positions belong to previous batches. The error floor is approximated by  $\text{BER} \approx s_{\min} M p^{s_{\min}} / B$ , where  $M$  denotes the number of minimal stall patterns that can be assigned to a batch. For SCs,  $M$  is derived in [1, Sec. V-B]. We use similar arguments to derive  $M$  for BCs and half-BCs listed in Table 1.

	staircase	braided	half-braided
$B$	$a^2 = \frac{n_c^2}{4}$	$3b^2 = \frac{n_c^2}{3}$	$\frac{3b^2 - b}{2} = \frac{n_c^2 - n_c}{6}$
$R$	$2\frac{k_c}{n_c} - 1$ $= 0.9167$	$2\frac{k_c}{n_c} - 1$ $= 0.9167$	$2\frac{k_c - 1}{n_c - 1} - 1$ $= 0.9166$
$W / \ell$	8 / 8	6 / 8	6 / 16
$D$	1,036,800	1,036,800	517,680
dec.	row/column	row/column	all at once
$s_{\min}$	$(t+1)^2 = 16$	$(t+1)^2 = 16$	$\frac{(t+1)(t+2)}{2} = 10$
$M$	$\binom{a}{t+1} \left( \binom{2a}{t+1} - \binom{a}{t+1} \right)$	*	$\binom{2b}{t+2} - \binom{b}{t+2}$
	* $\left( \binom{2b}{t+1} - \binom{b}{t+1} \right)^2 + 2\binom{b}{t+1} \left( \binom{3b}{t+1} - \binom{2b}{t+1} \right)$		

Table 1: Parameters

### 3.2 Results and Discussion

Results are shown in Fig. 2 for the SC (red), BC (blue), and half-BC (green). DE shows a slight (asymptotic) performance advantage for the SC, although for the chosen parameters at finite lengths, the SC and BC perform virtually identical. The DE results for the BC and half-BC are roughly the same (this can be seen by considering (E.1) and  $\eta$  for the two codes). The simulated half-BC has worse performance compared to the SC and BC caused by a different scaling behavior at finite lengths due to the reduced number of bits within the decoding window. Furthermore, the error floor is increased from  $\approx 10^{-20}$  for the SC and BC to  $\approx 10^{-14}$  due to the reduction of  $s_{\min}$ . On the other hand, the half-BC operates at only half the decoding delay. One may therefore improve the half-BC by employing a longer BCH code with parameters (960, 920, 4). This leaves the code rate unchanged, but significantly reduces the error floor to  $\approx 10^{-23}$  (note that  $s_{\min} = 15$ ) and also improves the waterfall performance, as predicted by DE and confirmed by the simulations (shown in brown). The delay is increased to  $D = 920,640$  bits, which is slightly less compared to the SC and BC.

Since the error floors were beyond the reach of software simulations, the inlet figure shows additional results for a half-BC employing a (600, 580, 2)-BCH code,  $W = 5$ , and  $\ell = 10$  to verify the analysis based on minimal stall patterns.

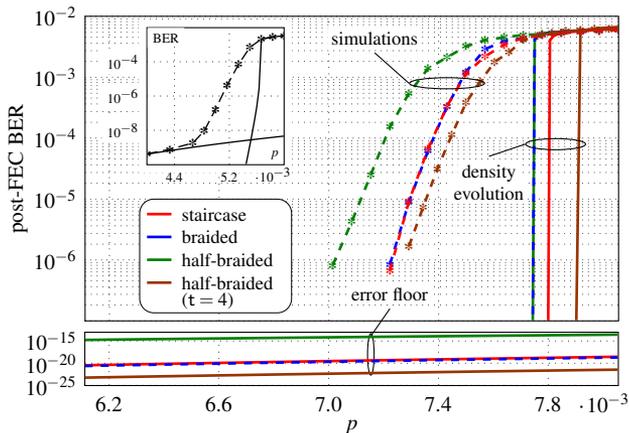


Figure 2: Results

## 4 Conclusion

We presented a DE analysis for deterministic GPCs, which applies to several code classes proposed for optical communications. DE can be used for a variety of different applications, e.g., parameter tuning, optimization of decoding schedules, or the design of new GPCs. As a case study, we compared SCs, BCs, and half-BCs. SCs and BCs perform similarly, while half-BCs can outperform both SCs and BCs at a lower error floor and decoding delay.

## References

- [1] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase codes: FEC for 100 Gb/s OTN,” *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.
- [2] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Mazahreh, “Iterative hard-decision decoding of braided BCH codes for high-speed optical communication,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.
- [3] J. Justesen and T. Høholdt, “Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes,” in *Proc. IEEE Information Theory Workshop (ITW)*, Tahoe City, CA, 2007.
- [4] J. Justesen, “Performance of product codes and related structures with iterated decoding,” *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.
- [5] C. Häger, et al., “Density evolution for deterministic generalized product codes on the binary erasure channel,” submitted to *IEEE Trans. Inf. Theory*, [available online] arxiv.org (2015).

- [6] A. J. Feltström, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, “Braided block codes,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jul. 2009.
- [7] H. D. Pfister, S. K. Emmadi, and K. Narayanan, “Symmetric product codes,” in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, 2015.



PAPER **F**

**Deterministic and Ensemble-Based Spatially-Coupled Product Codes**

Christian Häger, Henry D. Pfister, Alexandre Graell i Amat, and Fredrik Brännström,

in Proc. *IEEE Int. Symp. Information Theory (ISIT)*,  
Barcelona, Spain, 2016

*The layout has been revised.*

## Abstract

Several authors have proposed spatially-coupled (or convolutional-like) variants of product codes (PCs). In this paper, we focus on a parametrized family of generalized PCs that recovers some of these codes (e.g., staircase and block-wise braided codes) as special cases and study the iterative decoding performance over the binary erasure channel. Even though our code construction is deterministic (and not based on a randomized ensemble), we show that it is still possible to rigorously derive the density evolution (DE) equations that govern the asymptotic performance. The obtained DE equations are then compared to those for a related spatially-coupled PC ensemble. In particular, we show that there exists a family of (deterministic) braided codes that follows the same DE equation as the ensemble, for any spatial length and coupling width.

## 1 Introduction

Several authors have proposed modifications of the classical product code (PC) construction by Elias [1], typically by considering nonrectangular code arrays. These modifications can be regarded as generalized low-density parity-check (LDPC) codes [2], where the underlying Tanner graph consists exclusively of degree-2 variable nodes (VNs). We refer to such codes as generalized PCs (GPCs). For example, GPCs have been investigated by many authors as practical solutions for high-speed fiber-optical communications [3–7].

For the binary erasure channel (BEC), we are interested in the asymptotic iterative decoding performance of GPCs whose associated code arrays have a spatially-coupled or convolutional-like structure. Examples include braided codes [5,8] and staircase codes [4]. Spatially-coupled codes have attracted significant attention in the literature due to their outstanding performance under iterative decoding [9,10].

An asymptotic analysis is typically based on density evolution (DE) [11,12] using an ensemble argument. This approach was taken for spatially-coupled PCs in [13,14]. However, a randomly chosen code from these ensembles is unlikely to have a product array (row-column) structure and hence does not resemble the codes that are implemented in practice, e.g., staircase or braided codes. It is thus desirable to make precise statements about the performance of sequences of deterministic (and structured) GPCs.

We consider the high-rate regime, where one assumes that component codes correct a fixed number of erasures and then studies the case where the component code length  $n$  tends to infinity. Using a Chernoff bound, one finds that for any fixed erasure probability  $p$ , the decoding will fail for large  $n$  with high probability. Therefore, it is customary to let the erasure probability decay slowly as  $c/n$  for some  $c > 0$ . This leads to rigorous

decoding thresholds in terms of  $c$  which may be interpreted as the effective channel quality. The high-rate regime is also the regime that is relevant in practice: It is at high rates where GPCs are competitive compared to LDPC codes and practical GPCs typically employ long component codes with small erasure-correcting capability [3–5].

The main contribution of this paper is to show that, analogous to DE for code ensembles, there exists a large class of deterministic GPCs whose asymptotic performance in the high-rate regime is rigorously characterized in terms of a recursive DE equation. The code construction we propose here is sufficiently general to recover (block-wise) braided and staircase codes as special cases. Our result generalizes previous work in [3] from conventional PCs to a large class of deterministic GPCs. We further provide a detailed comparison between deterministic spatially-coupled PCs and the ensembles in [13, 14] via their respective DE equations. For example, we show that there exists a family of block-wise braided codes that follows the same DE recursion as the ensemble in [13]. This implies that certain ensemble-properties proved in [13] also apply to deterministic GPCs.

*Notation.* We use boldface letters to denote vectors and matrices (e.g.,  $\mathbf{x}$  and  $\mathbf{A}$ ). The symbols  $\mathbf{0}_m$  and  $\mathbf{1}_m$  denote the all-zero and all-one vectors of length  $m$ , where the subscript may be omitted. The tail-probability of a Poisson random variable is defined as  $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \Psi_{=i}(x)$ , where  $\Psi_{=i}(x) \triangleq \frac{x^i}{i!} e^{-x}$ . We use boldface to denote the element-wise application of a scalar-valued function to a vector, e.g.,  $\Psi_{\geq t}(\mathbf{x})$  applies  $\Psi_{\geq t}(\cdot)$  to each element in  $\mathbf{x}$ . For vectors, we use  $\mathbf{x} \succeq \mathbf{y}$  if  $x_i \geq y_i$  for all  $i$ . We define  $[m] \triangleq \{1, 2, \dots, m\}$ . The indicator function is denoted by  $\mathbb{1}\{\cdot\}$ .

## 2 Code Construction and Density Evolution for Deterministic Generalized Product Codes

We denote a GPC by  $\mathcal{C}_n(\boldsymbol{\eta})$ , where  $n$  is proportional to the number of constraint nodes (CNs) in the underlying Tanner graph and  $\boldsymbol{\eta}$  is a binary, symmetric  $L \times L$  matrix that defines the graph connectivity. Recall that GPCs also have a natural array representation: The code  $\mathcal{C}_n(\boldsymbol{\eta})$  can alternatively be defined as the set of all code arrays of a given shape (see Fig. 1 for examples) such that each row and column is a codeword in some component code. Thus, one may alternatively think about  $\boldsymbol{\eta}$  as specifying the array shape. We will see in the following that different choices for  $\boldsymbol{\eta}$  recover well-known code classes.

### 2.1 Code Construction

Let  $\gamma > 0$  be some fixed and arbitrary constant such that  $d \triangleq \gamma n$  is an integer. To construct the Tanner graph that defines  $\mathcal{C}_n(\boldsymbol{\eta})$ , assume that there are  $L$  positions. Then, place  $d$  CNs at each position and connect each CN at position  $i$  to each CN at position  $j$  through a VN if and only if  $\eta_{i,j} = 1$ .

*Example 1.* A PC is obtained by choosing  $L = 2$  and  $\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ . The two positions correspond to “row” and “column” codes. If we choose  $\gamma = 1$ , then the code array is of size  $n \times n$ .  $\triangle$

For a fixed  $n$ , the constant  $\gamma$  scales the number of CNs in the graph. This is inconsequential for the asymptotic analysis (where  $n \rightarrow \infty$ ) and  $\gamma$  manifests itself in the DE equations merely as a scaling parameter. Its choice will become clear once we discuss the comparison of codes defined by different  $\boldsymbol{\eta}$ -matrices in Sec. 3.1.

CNs at position  $i$  have degree  $d \sum_{j \neq i} \eta_{i,j} + \eta_{i,i}(d - 1)$ , where the second term arises from the fact that we cannot connect a CN to itself if  $\eta_{i,i} = 1$ . The CN degree specifies the length of the component code associated with the CN. We assume that each CN corresponds to a  $t$ -erasure correcting component code. This assumption is relaxed in Sec. 5.

## 2.2 Iterative Decoding

Suppose that a codeword of  $\mathcal{C}_n(\boldsymbol{\eta})$  is transmitted over the BEC<sup>1</sup> with erasure probability  $p = c/n$  for  $c > 0$ . The decoding is performed iteratively assuming  $\ell$  iterations of bounded-distance decoding for the component codes associated with all CNs. Thus, in each iteration, if the weight of an erasure pattern for a CN is less than or equal to  $t$ , the pattern is corrected. If the weight exceeds  $t$ , we say that the component code declares a decoding failure in that iteration.

## 2.3 Density Evolution

We wish to characterize the decoding performance in the limit  $n \rightarrow \infty$ . To that end, assume that we compute

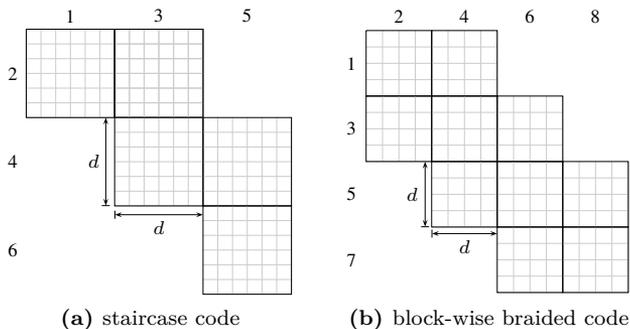
$$\mathbf{z}^{(\ell)} = \Psi_{\geq t+1}(c\mathbf{B}\mathbf{x}^{(\ell-1)}), \text{ with } \mathbf{x}^{(\ell)} = \Psi_{\geq t}(c\mathbf{B}\mathbf{x}^{(\ell-1)}), \quad (\text{F.1})$$

where  $\mathbf{x}^{(0)} = \mathbf{1}_L$  and  $\mathbf{B} \triangleq \gamma\boldsymbol{\eta}$ . The main result is as follows.

**Theorem 1.** *Let the random variable  $W$  be the fraction of component codes that declare decoding failures in iteration  $\ell$ . Then,  $W$  converges almost surely to  $\frac{1}{L} \sum_{i=1}^L z_i^{(\ell)}$  as  $n \rightarrow \infty$ .*

*Proof (Outline).* The decoding can be represented by applying a peeling algorithm to the residual graph which is obtained from the Tanner graph by deleting known VNs and collapsing erased VNs into edges [3, 5, 14]. Our code construction is such that the residual graph corresponds to an inhomogeneous random graph [15]. The expected value of a suitably defined function applied to such a graph converges to the expected value

<sup>1</sup>In practice, GPCs are used to correct errors and not erasures. However, the (rigorous) analysis over the BEC can be used to closely approximate the performance also over the binary symmetric channel, see, e.g., [3].



**Figure 1:** Code arrays for  $\mathcal{C}_{12}(\eta)$ , where in (a)  $\gamma = 1/2$  and in (b)  $\gamma = 1/3$ . Numbers indicate the position indices in the code construction.

of the same function applied to a multi-type Poisson branching process [15]. One can show that the peeling constitutes a valid function and that  $\frac{1}{L} \sum_{i=1}^L z_i^{(\ell)}$  corresponds to its expected value on the branching process. Concentration is established by applying the method of typical bounded differences [16]. For a complete proof we refer the reader to [17].  $\square$

Th. 1 is analogous to the DE analysis for LDPC codes [12, Th. 2]. For notational convenience, we define  $h(x) \triangleq \Psi_{\geq t}(cx)$ , so that the recursion in (F.1) can be written as

$$\mathbf{x}^{(\ell)} = \mathbf{h}(\mathbf{B}\mathbf{x}^{(\ell-1)}). \quad (\text{F.2})$$

**Definition 1.** The decoding threshold is defined to be

$$\bar{c} \triangleq \sup\{c \geq 0 \mid \mathbf{x}^{(\infty)} = \mathbf{0}_L\}. \quad (\text{F.3})$$

## 3 Spatially-Coupled Product Codes

### 3.1 Deterministic Spatially-Coupled Product Codes

We are interested in cases where  $\boldsymbol{\eta}$  (and hence  $\mathbf{B}$ ) has a band-diagonal “convolutional-like” structure. The associated code can then be classified as a spatially-coupled PC.

*Example 2.* For  $L \geq 2$ , the matrix  $\boldsymbol{\eta}$  describing a staircase code [4] has entries  $\eta_{i,i+1} = \eta_{i+1,i} = 1$  for  $i \in [L-1]$  and zeros elsewhere. The corresponding code array is shown in Fig. 1(a), where  $L = 6$ ,  $n = 12$ , and  $\gamma = 1/2$ .  $\triangle$

*Example 3.* For even  $L \geq 4$ , let  $\eta_{i,i+1} = \eta_{i+1,i} = 1$  for  $i \in [L-1]$ ,  $\eta_{2i-1,2i+2} = \eta_{2i+2,2i-1} = 1$  for  $i \in [L/2-1]$ , and zeros elsewhere. The resulting matrix  $\boldsymbol{\eta}$  describes a particular

instance of a block-wise braided code<sup>2</sup> [8]. The code array is shown in Fig. 1(b), where  $L = 8$ ,  $n = 12$ , and  $\gamma = 1/3$ .  $\triangle$

The threshold  $\bar{c}$  in Def. 1 is a function of  $\boldsymbol{\eta}$  and the scaling parameter  $\gamma$ . A reasonable scaling to compare different spatially-coupled PCs is to choose  $\gamma$  such that  $\lim_{L \rightarrow \infty} \frac{1}{L} \sum_{i=1}^L \sum_{j=1}^L B_{i,j} = 1$ . For example,  $\gamma = 1/2$  and  $\gamma = 1/3$  for staircase and braided codes, respectively. This ensures that in both cases the component codes have length  $n$ , except at the array boundaries, see Fig. 1. The matrix  $\mathbf{B}$  is then referred to as an averaging matrix.

### 3.2 Spatially-Coupled Product Code Ensembles

We wish to compare the obtained DE recursion in (F.2) to the DE recursion for the spatially-coupled PC ensemble defined in [13]. We review the necessary background in this section.

Let  $\mathcal{B}$  be a  $t$ -erasure correcting component code of length  $n$ . The Tanner graph corresponding to one particular code in the spatially-coupled  $(\mathcal{B}, m, L, w)$  ensemble, where  $L$  and  $w$  are referred to as the spatial length and coupling width, respectively, is constructed as follows (cf. [13, Def. 2]). Place  $m$  degree- $n$  CNs corresponding to  $\mathcal{B}$  at each position  $i \in [L]$  and place  $mn/2$  degree-2 VNs at each position  $i \in [L']$ , where  $L' \triangleq L - w + 1$ . The  $mn$  VN and CN sockets at each position are partitioned into  $w$  groups of  $mn/w$  sockets via a uniform random permutation. Let  $\mathcal{S}_{i,j}^{(v)}$  and  $\mathcal{S}_{i,j}^{(c)}$  be, respectively, the  $j$ -th group for the VNs and CNs at position  $i$ , where  $j \in [w]$ . The Tanner graph is constructed by connecting  $\mathcal{S}_{i,j}^{(v)}$  to  $\mathcal{S}_{i+j,w-j+1}^{(c)}$ .

The ensemble-averaged performance for  $m \rightarrow \infty$  is studied in [13]. Without going into the details, the obtained DE recursion in the high-rate regime (where, additionally,  $n \rightarrow \infty$  and  $p = c/n$ ) is given by [13, eq. (9)]

$$\tilde{\mathbf{x}}^{(\ell)} = c\mathbf{A}\Psi_{\geq t}(\mathbf{A}^\top \tilde{\mathbf{x}}^{(\ell-1)}), \quad (\text{F.4})$$

where  $\tilde{\mathbf{x}}^{(0)} = c\mathbf{1}_{L'}$  and  $\mathbf{A}$  is an  $L' \times L$  matrix with entries

$$A_{i,j} = w^{-1} \mathbb{1}\{1 \leq j - i + 1 \leq w\}, \text{ for } i \in [L'], j \in [L]. \quad (\text{F.5})$$

*Remark 1.* In [14], a modified spatially-coupled PC ensemble is considered. The obtained DE recursion is [14, eq. (4),  $v = 2$ ]

$$\mathbf{y}^{(\ell)} = c\mathbf{A}^\top \mathbf{A} \Psi_{\geq t}(\mathbf{y}^{(\ell-1)}), \quad (\text{F.6})$$

which is identical to (F.4) choosing  $\tilde{\mathbf{x}}^{(\ell)} = c\mathbf{A}\Psi_{\geq t}(\mathbf{y}^{(\ell)})$ .

<sup>2</sup>We are somewhat liberal in our interpretation of the definition in [8] which is based on multiple block permutators. In [8], these permutators are linked to the dimension of the component code, which turns out to be unnecessarily narrow for our purposes.

Observe that (F.4) exhibits a double averaging due to the randomized edge connections for both VNs and CNs at each position. Using the substitution  $\mathbf{x}^{(\ell)} = \Psi_{\geq t}(\mathbf{A}\tilde{\mathbf{x}}^{(\ell-1)})$  with  $\tilde{\mathbf{x}}^{(\ell)} = c\mathbf{A}\Psi_{\geq t}(\mathbf{x}^{(\ell)})$ , the recursion becomes

$$\mathbf{x}^{(\ell)} = \Psi_{\geq t}(c\tilde{\mathbf{B}}\mathbf{x}^{(\ell-1)}) = \mathbf{h}(\tilde{\mathbf{B}}\mathbf{x}^{(\ell-1)}), \quad (\text{F.7})$$

where  $\mathbf{x}^{(0)} = \mathbf{1}_L$  and  $\tilde{\mathbf{B}} \triangleq \mathbf{A}^\top \mathbf{A}$  is a symmetric  $L \times L$  matrix. For  $L = 6$ , the  $\tilde{\mathbf{B}}$ -matrices for  $w = 2$  and  $w = 3$  are, respectively, given by

$$\frac{1}{4} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (\text{F.8})$$

## 4 Comparison of Deterministic and Ensemble-Based Codes

Comparing the equations, one finds that the ensemble DE recursion (F.7) has the same form as (F.2). The difference lies in the averaging due to the matrix  $\tilde{\mathbf{B}}$ .

*Example 4.* It can be shown that staircase codes are contained in the ensemble for  $m = n/2$  and  $w = 2$  using a proper choice of permutations. It is therefore tempting to conjecture that for  $w = 2$ , the recursion (F.7) also applies to staircase codes. However, for staircase codes with  $L = 6$ , we have

$$\mathbf{B} = \gamma\boldsymbol{\eta} = \frac{1}{2} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (\text{F.9})$$

which is different from the matrix  $\tilde{\mathbf{B}}$  for  $w = 2$  in (F.8). △

*Example 5.* For the braided codes in Ex. 3, one can simplify (F.2) by exploiting the inherent symmetry in the code construction, which implies  $x_i^{(\ell)} = x_{i+1}^{(\ell)}$  for odd  $i$  and any  $\ell$ . It is then sufficient to retain odd (or even) positions in (F.2). With this simplification,

the effective averaging matrix<sup>3</sup> for  $L = 12$  is

$$\mathbf{B}' = \frac{1}{3} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad (\text{F.10})$$

where  $\mathbf{B}'$  may be used to replace  $\mathbf{B}$  in (F.2). Again, one finds that  $\mathbf{B}'$  is different from the matrices  $\tilde{\mathbf{B}}$  in (F.8).  $\triangle$

### 4.1 Ensemble Performance via Deterministic Codes

Since  $\boldsymbol{\eta}$  is binary, all entries in  $\mathbf{B}$  are either zero or equal to  $\gamma$ . To construct spatially-coupled PCs that follow the same DE recursion as the ensemble, we need to “emulate” different multiplicities in the matrix  $\mathbf{B}$ . This is done as follows.

**Definition 2.** For given  $L$  and  $w$ , let  $\gamma = 1/w^2$  and  $\mathbf{P} = w^2 \mathbf{A}^\top \mathbf{A}$ , where  $\mathbf{A}$  is defined by (F.5). We define  $\boldsymbol{\eta}$  as follows. First, replace each entry  $P_{i,j}$  in  $\mathbf{P}$  by a symmetric  $w \times w$  matrix with  $P_{i,j}$  ones in each row and column. The resulting  $wL \times wL$  matrix is denoted by  $\boldsymbol{\eta}'$ . Finally,  $\boldsymbol{\eta}$  is given by

$$\eta_{2i,2j-1} = \eta'_{i,j}, \quad \eta_{2i-1,2j} = \eta'_{j,i}, \quad \text{for } i, j \in [wL]. \quad (\text{F.11})$$

*Example 6.* Fig. 2 shows the (not necessarily unique) code array for  $L = 6$  and  $w = 3$ , where  $\mathbf{A}^\top \mathbf{A}$  is given in (F.8), which can be regarded as a type of braided code.  $\triangle$

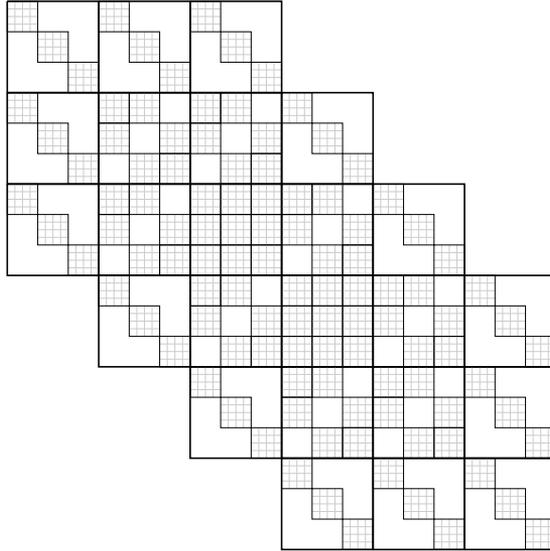
Using the structure of  $\boldsymbol{\eta}$  in Def. 2, one can show that the DE recursion for  $\mathcal{C}_n(\boldsymbol{\eta})$  in (F.2) is equivalent to (F.7). For example, the step in (F.11) is the opposite of the simplification in Ex. 5. The recursion defined by (F.7) constitutes an (unconditionally stable) scalar admissible system as defined in [10]. One may thus use the potential function approach in [10] to calculate decoding thresholds as follows (see also [5, 14]).

**Definition 3.** The single system potential function is defined as  $V_s(x) \triangleq \frac{1}{2}x^2 - H(x)$ , where  $H(x) = \int_0^x h(z) dz$ . In order to highlight the dependence of the potential function on the channel quality parameter  $c$ , we write  $V_s(x; c)$ .

**Definition 4.** The potential threshold is defined as

$$\bar{c}_p = \sup\{c \geq 0 \mid \min_{x \in [0;1]} V_s(x; c) \geq 0\}. \quad (\text{F.12})$$

<sup>3</sup>The reader may wonder to what code the matrix (F.10) corresponds to, i.e., the code  $\mathcal{C}_n(\boldsymbol{\eta})$  that results from using  $\boldsymbol{\eta} = 3\mathbf{B}'$ . One can show that  $\mathcal{C}_n(\boldsymbol{\eta})$  can be interpreted as a symmetric subcode of the braided code, see [17, 18].



**Figure 2:** Code array corresponding to  $\mathcal{C}_{24}(\boldsymbol{\eta})$  in Def. 2 with  $L = 6$ ,  $w = 3$ .

Using [10, Lem. 36], we have the following theorem.

**Theorem 2.** *Let  $\boldsymbol{\eta}$  and  $\gamma$  be as in Def. 2. For any  $c < \bar{c}_p$ , there exists  $w_0 < \infty$  such that for all  $w \geq w_0$  and all  $L$ , the DE recursion (F.2) for  $\mathcal{C}_n(\boldsymbol{\eta})$  converges to the  $\mathbf{0}$  vector.*

*Remark 2.* From Th. 2, the threshold of  $\mathcal{C}_n(\boldsymbol{\eta})$  satisfies  $\bar{c} \geq \bar{c}_p$  for all  $L$  and  $w$  sufficiently large. One can further show that  $\bar{c} \geq 2t - 2$  if, additionally,  $t$  is sufficiently large. The latter result was proved in [13, Lem. 8] for the spatially-coupled ensemble. It also applies to the deterministic braided codes in Def. 2, since the DE equations are equivalent.

## 4.2 Simpler Deterministic Codes

The curious structure of the code array in Fig. 2 is due to our attempt of “reverse-engineering” the DE equations of the ensemble by means of the deterministic code construction. This begs the question whether there exist other deterministic spatially-coupled PCs that exhibit a simpler structure but still achieve performance guarantees similar to Th. 2. The most natural candidate appears to be the extension of the block-wise braided code in Fig. 1(b) to larger coupling widths.

**Definition 5.** For given  $L$  and  $w$ , let  $\gamma = (2w - 1)^{-1}$  and let the  $L \times L$  matrix  $\boldsymbol{\eta}'$  be defined by  $\eta'_{i,j} = \mathbb{1}\{|i - j| < w\}$ . Finally, let  $\boldsymbol{\eta}$  be as in (F.11) for  $i, j \in [L]$ .

*Example 7.* For  $w = 2$ ,  $\boldsymbol{\eta}$  in Def. 5 recovers  $\boldsymbol{\eta}$  in Ex. 3. △

The resulting DE recursion for  $\mathcal{C}_n(\boldsymbol{\eta})$  is neither equivalent to the ensemble DE recursion nor to the recursion studied in [10]. However, one can still show the following.

**Theorem 3.** Let  $\boldsymbol{\eta}$  and  $\gamma$  be as in Def. 5. For any  $c < \bar{c}_p$ , there exists  $w_0 < \infty$  such that for all  $w \geq w_0$  and all  $L$ , the DE recursion (F.2) for  $\mathcal{C}_n(\boldsymbol{\eta})$  converges to the  $\mathbf{0}$  vector.

*Proof.* See the Appendix.  $\square$

## 5 Potential Threshold Optimization

In this section, we consider the case where we assign different erasure-correcting capabilities to the component codes. To that end, let  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_{t_{\max}})^\top$  be a probability vector (i.e.,  $\mathbf{1}^\top \boldsymbol{\tau} = 1$  and  $\boldsymbol{\tau} \succeq 0$ ), where  $\tau_t$  denotes the fraction of CNs at each position that can correct  $t$  erasures and  $t_{\max}$  is the maximum erasure-correcting capability. We further define the average erasure-correcting capability as  $\bar{t} \triangleq \sum_{t=1}^{t_{\max}} t \tau_t$ . The assignment can be done either deterministically if  $\tau_t d$  is an integer for all  $t$ , or independently at random according to  $\boldsymbol{\tau}$ . In both cases, the distribution  $\boldsymbol{\tau}$  manifests itself in the DE equation (F.2) by changing the function  $h$  defined in Sec. 2.3 to  $h(x) = \sum_{t=1}^{t_{\max}} \tau_t \Psi_{\geq t}(cx)$  (see [17] for details). This affects the potential function in Def. 3 and thus also the potential threshold in Def. 4. In particular, both quantities now depend on  $\boldsymbol{\tau}$  and this change is reflected in our notation by writing  $V_s(x; c, \boldsymbol{\tau})$  and  $\bar{c}_p(\boldsymbol{\tau})$ , respectively.

**Definition 6.** A distribution is said to be regular if  $\tau_{\bar{t}} = 1$  for  $\bar{t} \in \mathbb{N}$  and semi-regular if  $\tau_{\lfloor \bar{t} \rfloor} = 1 + \lfloor \bar{t} \rfloor - \bar{t}$  and  $\tau_{\lfloor \bar{t} \rfloor + 1} = \bar{t} - \lfloor \bar{t} \rfloor$  for  $\bar{t} \notin \mathbb{N}$ .

**Theorem 4.** For any fixed mean erasure-correcting capability  $\bar{t} \geq 2$ , a (semi-)regular distribution maximizes the potential threshold  $\bar{c}_p(\boldsymbol{\tau})$ .

*Proof.* See the Appendix.  $\square$

Th. 4 is in contrast to conventional PCs which typically benefit from employing component codes with different strengths. However, Th. 4 does not necessarily imply that there can be no practical value in employing different component codes also for spatially-coupled PCs. In practice, quantities such as the coupling width, the component code length, and the number of decoding iterations are constrained to be finite. Depending on the severity of these constraints, the potential threshold may not be a good performance indicator.

## 6 Conclusion

We studied the asymptotic performance of deterministic spatially-coupled PCs under iterative decoding. We showed that there exists a family of deterministic braided codes that performs asymptotically equivalent to a previously considered spatially-coupled PC ensemble. There also exists a related but structurally simpler braided code family that attains essentially the same asymptotic performance. Lastly, we showed that employing component code mixtures for spatially-coupled PCs is not beneficial from an asymptotic point of view.

## A Proof of Theorems 3 and 4

*Proof of Theorem 3.* The recursion of interest (after removing odd positions due to symmetry as explained in Ex. 5) is given by  $\mathbf{x}^{(\ell)} = \mathbf{h}(\mathbf{B}'\mathbf{x}^{(\ell-1)})$ , where  $\mathbf{B}' = \gamma\boldsymbol{\eta}'$  and  $\gamma, \boldsymbol{\eta}'$  are as in Def. 5. The authors in [10] study the recursion

$$\mathbf{y}^{(\ell)} = \mathbf{A}^\top \mathbf{f}(\mathbf{A}\mathbf{g}(\mathbf{y}^{(\ell-1)})) = \mathbf{A}^\top \mathbf{f}(\tilde{\mathbf{y}}^{(\ell)}) \quad (\text{F.13})$$

for suitable functions  $f, g$ , where  $\tilde{\mathbf{y}}^{(\ell)} = \mathbf{A}\mathbf{g}(\mathbf{y}^{(\ell-1)})$  is defined implicitly. Since  $h$  is strictly increasing and analytic, we can let both  $f = h$  and  $g = h$ . For this proof,  $\mathbf{A}$  is assumed to be of size  $L \times L + \tilde{w} - 1$  with  $A_{i,j} = \tilde{w}^{-1} \mathbb{1}\{1 \leq j - i + 1 \leq \tilde{w}\}$  for  $i \in [L], j \in [L + \tilde{w} - 1]$ , where  $\tilde{w} \triangleq 2w - 1$ . The potential function  $U_s(x; c) = h(x)x - H(x) - H(h(x))$  associated with the scalar recursion  $x^{(\ell)} = h(h(x^{(\ell-1)}))$  as defined in [10, eq. (4)] predicts the same potential threshold as the one in Def. 3. According to [10, Lem. 36], the claim in the theorem is thus true for the recursion (F.13). To show that it must also be true for the recursion of interest, we argue as follows. Assume that we swap the application of  $\mathbf{h}$  and  $\mathbf{B}'$  in the recursion of interest and then consider “two iterations at once” according to

$$\mathbf{z}^{(\ell)} = \mathbf{B}'\mathbf{h}(\mathbf{B}'\mathbf{h}(\mathbf{z}^{(\ell-1)})) = \mathbf{B}'\mathbf{h}(\tilde{\mathbf{z}}^{(\ell)}). \quad (\text{F.14})$$

We claim that (F.13) dominates (F.14), in the sense that  $\mathbf{y}^{(\infty)} = \mathbf{0}$  implies  $\mathbf{z}^{(\infty)} = \mathbf{0}$  (and thus  $\mathbf{x}^{(\infty)} = \mathbf{0}$ ). To see this, observe that  $\mathbf{y}^{(\ell)}$  has length  $L + \tilde{w} - 1$ , whereas  $\tilde{\mathbf{y}}^{(\ell)}, \mathbf{z}^{(\ell)}$ , and  $\tilde{\mathbf{z}}^{(\ell)}$  have length  $L$ . We use  $\mathbf{y}^{(\ell)} = ((\mathbf{y}_t^{(\ell)})^\top, (\mathbf{y}_c^{(\ell)})^\top, (\mathbf{y}_b^{(\ell)})^\top)^\top$  to denote the  $w - 1$  top,  $L$  center, and  $w - 1$  bottom entries in  $\mathbf{y}^{(\ell)}$ . We want to show that  $\mathbf{y}_c^{(\ell)} \succeq \mathbf{z}^{(\ell)}$  for all  $\ell$ . Assume this is true for  $\ell - 1$ . This gives the second inequality in

$$\tilde{\mathbf{y}}^{(\ell)} = \mathbf{A}\mathbf{h}(\mathbf{y}^{(\ell-1)}) \succeq \mathbf{B}'\mathbf{h}(\mathbf{y}_c^{(\ell-1)}) \succeq \mathbf{B}'\mathbf{h}(\mathbf{z}^{(\ell-1)}) = \tilde{\mathbf{z}}^{(\ell)},$$

where the first inequality follows from  $\mathbf{y}_t^{(\ell-1)}, \mathbf{y}_b^{(\ell-1)} \succeq \mathbf{0}$  (since  $\mathbf{y}^{(\ell)} \succeq \mathbf{0}$  for all  $\ell$ ) and the (almost identical) structure of  $\mathbf{A}$  and  $\mathbf{B}'$ . Observe that we have  $\mathbf{y}_c^{(\ell)} = \mathbf{B}'\mathbf{h}(\tilde{\mathbf{y}}^{(\ell)})$ . Also  $\mathbf{z}^{(\ell)} = \mathbf{B}'\mathbf{h}(\tilde{\mathbf{z}}^{(\ell)})$  and, since we have just shown that  $\tilde{\mathbf{y}}^{(\ell)} \succeq \tilde{\mathbf{z}}^{(\ell)}$ , the claim follows by induction on  $\ell$ .  $\square$

*Proof of Theorem 4.* Using integration by parts, one may verify that the potential function in Def. 3 is given by

$$V_s(x; c, \boldsymbol{\tau}) = x^2/2 - x + (\bar{t} - \mathcal{L}_{\boldsymbol{\tau}}(cx))/c, \quad (\text{F.15})$$

where we defined  $\mathcal{L}_{\boldsymbol{\tau}}(x) \triangleq \sum_{t=1}^{\bar{t}} \tau_t \mathcal{L}(t, x)$ , with  $\mathcal{L}(t, x) \triangleq \sum_{k=0}^{t-1} \Psi_{=k}(x)(t - k)$  for  $t \in \mathbb{N}$ . For any fixed  $x \geq 0$ , we also define the affine extension of  $\mathcal{L}(t, x)$  for  $t \in [1, \infty)$  as

$$\mathcal{L}(t, x) = \mathcal{L}(\lfloor t \rfloor, x) + (\mathcal{L}(\lceil t \rceil, x) - \mathcal{L}(\lfloor t \rfloor, x))(t - \lfloor t \rfloor). \quad (\text{F.16})$$

The proof relies on the fact that  $\mathcal{L}(t, x)$  is convex in  $t \in [1, \infty)$  for any  $x \geq 0$ . Indeed, since  $\mathcal{L}(t, x)$  is the affine extension of a discrete function, it suffices to show that for  $t \in \{2, 3, \dots\}$ ,

$$\mathcal{L}(t-1, x) + \mathcal{L}(t+1, x) = 2\mathcal{L}(t, x) + \Psi_{=t}(x) \tag{F.17}$$

$$\geq 2\mathcal{L}(t, x), \tag{F.18}$$

since  $\Psi_{=t}(x) \geq 0$  with equality if and only if  $x = 0$ . As a consequence, for any distribution  $\boldsymbol{\tau}$  with average erasure-correcting capability  $\bar{t}$  and any  $x \geq 0$ , we have

$$\mathcal{L}_{\boldsymbol{\tau}}(x) \geq \mathcal{L}(\bar{t}, x) = \mathcal{L}_{\boldsymbol{\tau}_{\text{reg}}}(x), \tag{F.19}$$

where  $\boldsymbol{\tau}_{\text{reg}}$  denotes the (semi-)regular distribution in Def. 6.

Now, let  $\mathbf{t} \triangleq (1, 2, \dots, t_{\text{max}})^\top$  and consider

$$\max_{\boldsymbol{\tau} \in \mathcal{T}} \bar{c}_p(\boldsymbol{\tau}) \text{ subject to } \mathbf{t}^\top \boldsymbol{\tau} = \bar{t}, \tag{F.20}$$

where  $\mathcal{T} = \{\boldsymbol{\tau} \in \mathbb{R}^{t_{\text{max}}} \mid \mathbf{1}^\top \boldsymbol{\tau} = 1, \boldsymbol{\tau} \geq 0\}$ . This can be equivalently written in epigraph form as

$$\max_{c \geq 0, \boldsymbol{\tau} \in \mathcal{T}} c \text{ subject to } c \leq \bar{c}_p(\boldsymbol{\tau}), \mathbf{t}^\top \boldsymbol{\tau} = \bar{t}. \tag{F.21}$$

According to (F.12), the first constraint in (F.21) is equivalent to  $V_s(x; c, \boldsymbol{\tau}) \geq 0$  for  $x \in [0; 1]$ . Assume that (F.21) is maximized by some  $(c^*, \boldsymbol{\tau}^*)$ . Then, for all  $x \in [0; 1]$ , we have

$$0 \leq V_s(x; c^*, \boldsymbol{\tau}^*) \leq V_s(x; c^*, \boldsymbol{\tau}_{\text{reg}}), \tag{F.22}$$

where the last inequality follows from (F.15) and (F.19). Thus, the (semi-)regular distribution  $\boldsymbol{\tau}_{\text{reg}}$  is feasible and attains (at least) the same threshold value  $c^*$ .  $\square$

## References

- [1] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Apr. 1954.
- [2] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
- [3] J. Justesen, "Performance of product codes and related structures with iterated decoding," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.
- [4] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN," *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

- [5] Y.-Y. Jian, H. D. Pfister, K. R. Narayanan, R. Rao, and R. Mazahreh, “Iterative hard-decision decoding of braided BCH codes for high-speed optical communication,” in *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA, 2014.
- [6] L. M. Zhang and F. R. Kschischang, “Staircase codes with 6% to 33% overhead,” *J. Lightw. Technol.*, vol. 32, no. 10, pp. 1999–2002, May 2014.
- [7] C. Häger, A. Graell i Amat, H. D. Pfister, A. Alvarado, F. Brännström, and E. Agrell, “On parameter optimization for staircase codes,” in *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA, 2015.
- [8] A. J. Feltström, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, “Braided block codes,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jul. 2009.
- [9] S. Kudekar, T. Richardson, and R. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [10] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, “A simple proof of Maxwell saturation for coupled scalar recursions,” *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.
- [11] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, “Analysis of random processes via and-or tree evaluation,” in *Proc. 9th Annual ACM-SIAM Symp. Discrete Algorithms*, San Francisco, CA, 1998.
- [12] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [13] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high rates with iterative hard-decision decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, 2012.
- [14] L. M. Zhang, D. Truhachev, and F. R. Kschischang, “Spatially-coupled split-component codes with bounded-distance component decoding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Hong Kong, 2015.
- [15] B. Bollobás, S. Janson, and O. Riordan, “The phase transition in inhomogeneous random graphs,” *Random Structures and Algorithms*, vol. 31, no. 1, pp. 3–122, Aug. 2007.
- [16] L. Warnke, “On the method of typical bounded differences,” *Combinatorics, Probability and Computing (to appear)*, 2015. [Online]. Available: <http://arxiv.org/pdf/1212.5796.pdf>

- [17] C. Häger, H. D. Pfister, A. Graell i Amat, and F. Brännström, “Density evolution for deterministic generalized product codes on the binary erasure channel,” *submitted to IEEE Trans. Inf. Theory*, 2015. [Online]. Available: <http://arxiv.org/pdf/1512.00433.pdf>
- [18] H. D. Pfister, S. K. Emmadi, and K. Narayanan, “Symmetric product codes,” in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, 2015.