SPATIALLY COUPLED LDPC CODES AND
COOPERATIVE COMMUNICATION

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Lai Wei

__________________________

Thomas E. Fuja, Director

__________________________

Daniel J. Costello, Jr., Co-Director

Graduate Program in Electrical Engineering
Notre Dame, Indiana
April 2015
© Copyright by
Lai Wei
2014
All Rights Reserved
In this dissertation we focus on three topics in wireless communication: 1) design of spatially coupled low-density parity-check (SC-LDPC) codes over the finite field (i.e., the Galois field) of size $q$ (denoted as $\text{GF}(q)$) for windowed decoding (WD), 2) cooperative communication using binary LDPC block codes in the context of bandwidth-efficient modulation, and 3) the use of binary rate-compatible SC-LDPC codes in a binary coded cooperation system.

As for the first topic, we consider the generalization of binary SC-LDPC codes to $\text{GF}(q)$, $q \geq 2$, and discuss design rules for $q$-ary SC-LDPC code ensembles based on their iterative belief propagation (BP) decoding thresholds, with particular emphasis on low-latency WD. We consider transmission over both the binary erasure channel (BEC) and the binary-input additive white Gaussian noise channel (BIAWGNC), and present results for a variety of $(J,K)$-regular SC-LDPC code ensembles constructed over $\text{GF}(q)$ using protographs. Thresholds are calculated using protograph versions of $q$-ary density evolution (for the BEC) and $q$-ary extrinsic information transfer analysis (for the BIAWGNC). We show that WD of $q$-ary SC-LDPC codes provides significant threshold gains compared to corresponding (uncoupled) $q$-ary LDPC block code (LDPC-BC) ensembles when the window size $W$ is large enough; we also show that these gains increase as the finite field size $q = 2^m$ increases. Moreover, we
demonstrate that our design rules provide WD thresholds that are close to capacity, even when both \( m \) and \( W \) are relatively small (thereby reducing complexity and latency). Analysis shows that, compared to standard flooding-schedule decoding, WD of \( q \)-ary SC-LDPC code ensembles results in significant reductions in both decoding complexity and decoding latency, and that these reductions increase as \( m \) increases. For applications with a near-threshold performance requirement and a constraint on decoding latency, we show that using \( q \)-ary SC-LDPC code ensembles, with moderate \( q > 2 \) instead of their binary counterparts results in reduced decoding complexity.

Regarding the second topic, we consider a communication scenario in which a pair of cooperating partners (i.e., two source nodes) convey their data to a common destination. To mitigate fading, each partner transmits its own local data and acts as a relay for the other partner. Specifically, relaying is incorporated into the channel coding function: local data (originating at the transmitting node) and relayed data (originating at the partner of the transmitting node) are encoded separately, and the resulting bitstreams are then multiplexed together prior to bandwidth-efficient modulation (e.g., 8-PSK, 16-QAM, etc.) in such a way that the relayed coded bits partition the signal constellation into sparse subsets, as in Ungerboeck’s set partitioning approach to coded modulation. The partner, having knowledge of the relayed bits, is able to demodulate each block-faded and noise-corrupted symbol using the appropriate sparse sub-constellation, improving the partner-to-partner link. The destination benefits in two ways: indirectly from the increased diversity made possible by the enhanced partner-to-partner link, and directly by exploiting the set-partition labeling. Outage results and frame error rate simulations using low-density parity-check codes demonstrate substantial performance gain, fundamentally and practically, over the conventional “time-sharing” approach to cooperation.

Finally, for the third topic, we investigate the use of rate-compatible SC-LDPC codes for binary coded cooperation. In the same “two sources, one destination” model
as described in the second topic, one source node relays additional parity-check bits for its partner’s latest transmission to provide cooperative diversity at the destination. Different families of SC-LDPC code ensembles are generated by applying the edge spreading technique to several good rate-compatible protograph-based LDPC-BC ensembles reported in the literature. Simulation of the outage behavior shows that, using SC-LDPC code ensembles, system performance approaches the theoretical limit, regardless of whether the original uncoupled LDPC-BC ensembles were designed specifically for coded cooperation or not. The same result holds when WD is used to reduce decoding latency.
To my parents and my fiancée
CONTENTS

FIGURES ................................................................. vi
TABLES ................................................................. ix
ACKNOWLEDGMENTS ................................................. x

CHAPTER 1: BACKGROUND AND OUTLINE .......................... 1
1.1 Background: Channel Coding .................................. 2
1.2 Outline of the Dissertation .................................... 5

CHAPTER 2: DESIGN OF SPATIALLY COUPLED LDPC CODES
OVER GF(q) FOR WINDOWED DECODING ......................... 8
2.1 Introduction ....................................................... 8
2.2 Windowed Decoding of Protograph-Based q-ary SC-LDPC Code Ensembles ........................................... 13
  2.2.1 Protograph-based q-ary SC-LDPC Code Ensembles ......... 13
  2.2.2 Windowed Decoding (WD) ................................... 17
  2.2.3 Code Ensemble Construction ................................ 18
    2.2.3.1 (J,K)-Regular LDPC-BC Ensembles ..................... 18
    2.2.3.2 Edge Spreadings of B ................................... 19
    2.2.3.3 (J,K)-Regular SC-LDPC Code Ensembles ............... 20
  2.3 Threshold Analysis of q-ary SC-LDPC Code Ensembles on the BEC ..... 22
    2.3.1 Protograph q-ary Density Evolution (DE) on the BEC .... 22
      2.3.1.1 Protograph DE for q-ary LDPC Code Ensembles ....... 23
      2.3.1.2 Flooding-Schedule Decoding (FSD) Thresholds for q-ary SC-LDPC Code Ensembles ................. 26
      2.3.1.3 Windowed Decoding (WD) Thresholds for q-ary SC-LDPC Code Ensembles ................. 27
  2.3.2 Numerical Results: k = 2 (R = 1/2) ......................... 28
    2.3.2.1 The (2, 4) Ensembles .................................. 29
    2.3.2.2 The (3, 6) Ensembles .................................. 31
    2.3.2.3 The (4, 8) and (5, 10) Ensembles .......... 40
  2.3.3 Numerical Results: k = 3 and k = 4 (R > 1/2) .............. 47
    2.3.3.1 (J,K) = (3, 9), k = 3 .................................. 48
    2.3.3.2 (J,K) = (3, 12), k = 4 ................................ 51
2.4 Threshold Analysis of $q$-ary SC-LDPC Code Ensembles on the BIAWGNC ................................................. 53
  2.4.1 $q$-ary Protograph EXIT Analysis on the BIAWGNC .................................................. 53
  2.4.2 Numerical Results ................................................................. 54
2.5 Decoding Latency and Decoding Complexity ....................................................... 54
  2.5.1 Decoding Latency ................................................................. 56
  2.5.2 Decoding Complexity ............................................................. 57
  2.5.3 Numerical Results ................................................................. 58
    2.5.3.1 WD vs. FSD, with the Same Decoding Threshold .......... 58
    2.5.3.2 WD Complexity as a Function of $m$ and $W$, with
            Equal Latency .............................................................. 60
    2.5.3.3 $B(J, 2J, m)$ vs. $C_1(J, 2J, m, 3)$, with Equal Latency 64
2.6 Summary ................................................................. 64

CHAPTER 3: BANDWIDTH-EFFICIENT COOPERATIVE COMMUNICATION ......................................................... 67
3.1 Introduction ................................................................. 67
3.2 System Description and General Approaches ....................................................... 71
3.3 Cooperation via Set Partitioning and Repetition ....................................................... 73
  3.3.1 SP-R: Node A Transmitting .................................................. 74
  3.3.2 SP-R: Node B Receiving ..................................................... 77
  3.3.3 SP-R: Node D Receiving ..................................................... 78
  3.3.4 Simulation: Frame Error Rate at Destination ....................................................... 83
3.4 Set-Partitioning Coded Cooperation (SP-CC): Achieving Different Cooperation Levels ....................................................... 85
  3.4.1 A Multiplexer-Based Approach ............................................... 85
  3.4.2 A Mixing Approach ............................................................. 86
3.5 Outage Analysis of Cooperation Based on Set Partitioning ....................................................... 87
  3.5.1 SP-R Outage: At the Partner Nodes ........................................ 88
  3.5.2 SP-R Outage: At the Destination Node ........................................ 88
    3.5.2.1 State $S_0$ ................................................................. 89
    3.5.2.2 State $S_1$ ................................................................. 89
    3.5.2.3 State $S_2$ ................................................................. 91
    3.5.2.4 State $S_3$ ................................................................. 92
  3.5.3 Outage Analysis for Other Systems ........................................ 93
    3.5.3.1 Time-Sharing Repetition ................................................ 93
    3.5.3.2 Set-Partitioning Coded Cooperation ....................................................... 94
  3.5.4 Monte Carlo Simulation ............................................................. 95
    3.5.4.1 Comparing Two Repetition-Based Schemes: SP-R versus
            TS-R ................................................................. 96
    3.5.4.2 Comparison: TS-R, TS-CC, SP-R and SP-CC ............. 97
3.6 Two Variations on the Set-Partitioning Repetition Approach ....................................................... 98
  3.6.1 Turbo-Decoding Structure ..................................................... 98
  3.6.2 Frame Interleaving ............................................................. 101
2.1 A (3, 6)-regular protograph and its corresponding base-matrix representation. Black circles correspond to variable nodes and crossed boxes correspond to check nodes. ........................ 9

2.2 (a) A sequence of $L = 8$ uncoupled (3, 6)-regular LDPC-BC protographs, and (b)-(e) various (3, 6)-regular SC-LDPC protographs constructed following the edge-spreading procedure with the coupling length $L = 8$. ........................ 14

2.3 WD example with window size $W = 3$: at $t = 1$ (solid red), $t = 2$ (dotted blue), and $t = 3$ (dash green). $J = 3$, $K = 6$, $w = 1$; $B_0 = [1, 1]$ and $B_1 = [2, 2]$, both of size $(c-b) \times c = 1 \times 2$, for construction of $B_{SC}$ (2.2). For each window position/time instant, the first $c = 2$ column blocks are target symbols. ................... 16

2.4 FSD and WD thresholds of $C_1(2, 4, m)$. ......................... 30

2.5 FSD threshold comparison of the (3, 6, $m$) ensembles. ........... 32

2.6 FSD thresholds of SC-LDPC code ensembles with different coupling lengths $L$. ......................................................... 33

2.7 WD thresholds of the $C_2(3, 6, m)$ and $C_1(3, 6, m, 1)$ ensembles. FSD thresholds are included as benchmarks. ....................... 35

2.8 Continuing with Fig. 2.7: WD thresholds of the $C_1(3, 6, m, 2)$ and $C_1(3, 6, m, 3)$ ensembles. ................................. 36

2.9 The portion of the base matrix covered by the window when $W = 4$. 39

2.10 WD thresholds of $q$-ary SC-LDPC code ensembles with $w = 1$ and $W = 3$ from: (a) the (3, 6, $m$) ensembles, and (b) the (4, 8, $m$) ensembles. FSD thresholds of the corresponding $q$-ary LDPC-BC ensembles are included for reference. .......................... 41

2.11 Continuing with Fig 2.10: WD thresholds of $q$-ary SC-LDPC code ensembles with $w = 1$ and $W = 3$ from the (5, 10, $m$) ensembles. Moreover, the WD thresholds of the $C_1(J, 2J, m, 2)$ ensembles, $J = 3$, 4, and 5, with $W = 5$ are shown in (d). ......................... 42

2.12 Comparison of WD thresholds: type 1 spreading vs. type 2 spreading for $J = 3$ with $W = 5$, for $J = 4$ with $W = 5$, and for $J = 5$ with $W = 4$. Note that for the latter two comparisons, $C_1(J, 2J, m, 1)$ has better thresholds than $C_1(J, 2J, m, 2)$ for all $m$ .......... 45
2.13 WD thresholds of $C_{J-1}(J, 2J, m = 2)$ ensembles as $J$ increases for window sizes $W = 4, 6, 8,$ and 10. (For $C_{4}(5, 10, m)$ with $W = 4$, $\epsilon_{WD}^*(m, 4) = 0$, because the minimum required $W$ is 5.)  

2.14 WD thresholds of the $(3, 9, m)$ SC ensembles with $w = 1$: (a) $W = 4$, and (b) $W = 10$, a sufficiently large window size such that the best WD thresholds are achieved for all the SC-LDPC code ensembles. The FSD thresholds of $B(3, 9, m)$ are included as a benchmark.  

2.15 Percentage divergence of the best achievable WD thresholds $\hat{\epsilon}_{WD}(m)$ from the corresponding channel capacities for (a) the $C_{1}(3, 6, m, 3)$, $C_{1}(3, 9, m, 4)$, and $C_{1}(3, 12, m, 5)$ ensembles with all $E_{B}$ spreading, and (b) the $C_{1}(3, 6, m, 2)$, $C_{1}(3, 9, m, 3)$, and $C_{1}(3, 12, m, 4)$ ensembles containing only one $E_{A}$ spreading.  

2.16 FSD thresholds of the $(2, 4, m)$ and $(3, 6, m)$ ensembles and WD thresholds of $C_{1}(3, 6, m, 2)$ on the BLAVGNC.  

2.17 WD vs. FSD for $C_{1}(3, 6, m, 2)$: comparison of (a) decoding complexity, and (b) decoding latency.  

2.18 Comparison of decoding complexity: $B(3, 6, m)$ using FSD and $C_{1}(3, 6, m, 3)$ using WD with $W = 2$.  

3.1 Node A and B cooperate to convey their data to a common destination Node D.  

3.2 Transmitter in the SP-R approach: Node A operates in the cooperative transmission mode.  

3.3 The SP labeling for 16-QAM and 64-QAM modulation: those solid signal points have the same high-order half of the labeling.  

3.4 Receiver in the SP-R approach at Node B: decoder initialization is more reliable when the SP labeling is adopted.  

3.5 The SP-R approach: Node D decodes $i_{L}^{A}(t)$ in state $S_{1}$.  

3.6 The SP-R approach, FER comparison at Node D: 8-PSK, $\eta = 1$ bit/symbol, $k = 1500$.  

3.7 The SP-R approach, FER comparison at Node D: 64-QAM, $\eta = 1.5$ bits/symbol, $k = 1500$.  

3.8 Outage probabilities of the SP-R approach at the destination: 16-QAM, $\eta = 0.4, 0.8$ and 1.2 bits/symbol.  

3.9 Performance gain of SP-R over TS-R: $\eta$ vs. average $E_{b}/N_{0}$ (dB) when the outage probability is fixed at $10^{-2}$; 16-QAM.  

3.10 Comparing SP-R FER obtained using LDPC block codes with outage probability: 16-QAM, $\eta = 1.0$ bit/symbol, $k = 1000$.  

3.11 Outage probability of TS-R, TS-CC, SP-R and SP-CC: 16-QAM, $\eta = 1.0$ bit/symbol.
3.12 The turbo-SP-R approach, FER comparison at Node D: 16-QAM, $\eta = 0.5 \text{ bit/symbol}, k = 500$. .................................................. 100

3.13 The $d$-SP-R approach, FER comparison at Node D: 16-QAM, $\eta = 1.0 \text{ bit/symbol}, k = 1000$. .................................................. 103

4.1 Transmitted codeword bits of Node A and B in one time slot. ....... 108

4.2 Four possible states $S_i$, $i = 0, 1, 2, 3$ of Node D decoding Node A’s local data in coded cooperation, depending on transmission of Node A and B in the local phase and the relay phase. ......................... 110

4.3 Rate compatibility is achieved by graph extension in the RC-SC-LDPC code family; $w = 1$. Every base component matrix is extended, and the additional parity variable nodes are only connected to the additional check nodes. .................................................. 113

4.4 Thresholds of the $k = 11$ code ensemble in the RC-SC-PBRL code family in (a) state $S_1$ and (b) state $S_3$ at the destination Node D, compared with capacity of the BPSK modulation. Cooperation level $\beta = 1/2$, and spectral efficiency $\eta = 0.3278$. ......................... 126

4.5 Thresholds of the $k = 11$ code ensemble in the RC-SC-PBRL code family in (a) state $S_1$ and (b) state $S_3$ at the destination Node D, when spectral efficiency $\eta = 0.3278$. Cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18) .... 129

4.6 Comparison of the system outage probability at Node D: RC-PBRL, RC-PN-PBRL, RC-SC-PBRL, and RC-SC-PN-PBRL. Cooperation level $\beta = 1/2$. .................................................. 131

4.7 Comparison of the system outage probability at Node D, when RC-SC-PBRL is applied: cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18). Spectral efficiency $\eta = 0.3278$. .................................................. 134

4.8 Comparison of the system outage probability at Node D: RC-R-LDPC, RC-SC-R-LDPC with $m_s = 1$ and $L = 100$, and RC-SC-R-LDPC with $m_s = 2$ and $L = 200$. Cooperation level $\beta = 1/2$. ......................... 137
# TABLES

2.1 Decoding Complexity of $C_1(3, 6, m, 2)$ . . . . . . . . . . . . . . . . . . 62

2.2 Continuing with Table 2.1: Decoding Complexity of $C_1(3, 6, m, 2)$ . . 63

3.1 Four Possible States: Node D Decoding $i^A_L(t)$ . . . . . . . . . . . . . . 79

4.1 Thresholds Using the RC-SC-PBRL Code Family
(Measured in AWGN Symbol Energy-to-Noise Ratio $E_s/N_0$) . . . . . . 125

4.2 For Figure 4.6: Average SNR $E_b/N_0$ (dB)
When the System Outage Probability $P_O = 10^{-2}; \beta = 1/2$ . . . . . . 133

4.3 For Figure 4.7: RC-SC-R-LDPC, Average SNR $E_b/N_0$ (dB)
When the System Outage Probability $P_O = 10^{-2}; \eta = 0.3278$ . . . . . 135

4.4 Average $E_b/N_0$ (dB) When the System Outage Probability $P_O = 10^{-2}$:
RC-SC-PBRL with Windowed Decoding, $\beta = 1/2$, and $\eta = 0.3278$ . . 136

4.5 For Figure 4.8: Average $E_b/N_0$ (dB)
When the System Outage Probability $P_O = 10^{-2}; \beta = 1/2$ . . . . . . 138
ACKNOWLEDGMENTS

My advisor, Dr. Thomas E. Fuja, whom I have worked with since 2010, and my co-advisor, Dr. Daniel J. Costello, Jr., whom I have worked with since 2012, are the two persons whom I would like to thank most. When I was choosing my PhD research topic, a lot of friends and colleagues tried to convince me that “channel coding is already in the dead end” and “you have to follow the trends of the literature.” I was quite fortunate to meet Dr. Fuja, who encouraged me and guided me into the real world of channel coding behind all the illusions and misunderstandings and helped me to choose the starting point of my research. Afterwards, I gradually shifted my research focus from application to code design, and Dr. Fuja and Dr. Costello led me into cutting-edge topics in the area. The meetings and discussions with them were always enjoyable and rewarding, and their wisdom, experience, and leadership in channel coding always inspired me, both in my research and in my life. I spent five years at the University of Notre Dame in my twenties, and being a student of Dr. Fuja and Dr. Costello was the greatest part of the memory. Maybe many years in the future, I will forget all the details of my PhD research, but the training of independent thinking and the detailed-oriented style of work and life that I learned from Dr. Fuja and Dr. Costello will always benefit me in my personal development.

I would like to thank David G. M. Mitchell (research professor) and Kechao Huang (visiting scholar), who shared with me their smart ideas and excellent work in channel coding and offered tremendous help during the preparation of my publications. Also, thank Shashank Maiya and Lei Xiao, former members of Dr. Fuja and Dr. Costello’s group, who shared with me their research resources so that I was able to ramp up
quickly at the beginning.

I would like to thank Xinchen Zhang and Yaoyi Guan, my two roommates during my life in Notre Dame. They never said no when I needed their help; life would have been much tougher without them. Chatting with them about technology, history, and politics was always delightful. Also, thank Zhanwei Sun and Zhenhua Gong, who have been offering me great academic and career advice ever since I first arrived in Notre Dame. Thanks to Ding Nie, Jun Chen, and all the others Chinese folks in the Department of Electrical Engineering, University of Notre Dame.

I would like to thank Qualcomm Inc. and Mitsubishi Electric Research Laboratories for offering me the opportunities of internships. The internships broadened my view of how my research fits into industry, like cellular communication and optical communication, so that I can provide valuable feedbacks to my group in Notre Dame.

Finally, thanks to the Center for Research Computing (CRC) at the University of Notre Dame. Simulations in channel coding can be very time-consuming, and the computing resources provided by CRC guaranteed that I was able to verify designs in a timely manner.

Thank you all.
Overall, this dissertation fits into the structure of “channel coding: design and application.” As for channel coding, we focus on low-density parity-check (LDPC) codes, one of the most important milestones in the history of coding theory and one of the most competitive candidates for channel coding implementation in today’s numerous wired and wireless communication systems. Especially, we discuss and propose rules for the design of “universally-good” spatially-coupled LDPC (SC-LDPC) codes, which have aroused huge interest in both academia and industry in recent years due to their theoretical capacity-achieving performance on different types of channels. Finally, as for application, we use SC-LDPC codes and LDPC block codes in the context of cooperative communications, in which distributed transmitters cooperate and form a virtual antenna array to provide spatial diversity; we show how to exploit the coding gain besides the diversity gain to further improve the system performance, both theoretically (i.e., the outage probability) and practically (i.e., the frame error rate).

To avoid duplication, for a brief overview of SC-LDPC codes and cooperative communication, refer to the introductions in Chapter 2, 3, and 4. But since “channel coding” serves as the soul of the whole dissertation, in this chapter, we summarize some very high-level background information about it without throwing the readers into a large number of references; the discussion is mainly based on the author’s experience in both coding theory research and cellular communication industry. After this, we present the outline of the dissertation to show how everything is organized
around channel coding.

1.1 Background: Channel Coding

In his revolutionary work “A Mathematical Theory of Communication” [1], Claude Shannon stated that

*The fundamental problem of communication is that of reproducing at one point, either exactly or approximately, a message selected at another point.*

Over the decades, countless efforts have been dedicated to the goal of conveying signals reliably and efficiently through different types of “imperfect” wired and wireless channels with additive white Gaussian noise, path loss, shadowing, multipath fading, Doppler spread, etc. and among all the strategies, channel coding has always been one of the kernel techniques to combat channel corruptions to the signal.

The role of channel coding in communication reliability is quite straightforward. From the classical channel codes represented by Hamming codes and convolutional codes, to the modern channel codes represented by Turbo codes and low-density parity-check (LDPC) codes, the basic principle is consistent: by introducing structured redundancy in terms of “parity bits/symbols” during the encoding procedure at the transmitter side, the decoding procedure at the receiver side is able to detect and/or correct errors within the corresponding capability. We emphasize that the redundancy is *structured*, i.e., added on purpose, in contrast to the case of source coding in which the unstructured redundancy is reduced, if not completely removed, in order to represent the source data more efficiently. Clearly, channel coding has to balance between reliability and *efficiency*, and the groundbreaking work of trellis coded modulation (TCM) [2] shows that by jointly designing modulation and coding carefully, coded modulation with a higher modulation order is able to outperform uncoded modulation with a lower modulation order, when the bandwidth efficiency (bits per symbol) is the same. Besides TCM, another milestone in the history of
coded modulation is bit-interleaved coded modulation (BICM) [3]. In BICM, channel coding and modulation are separated by a pseudo-random interleaver between them; clearly, to some extent, the design is simplified compared to TCM, and more importantly, in the case of a slow-fading channel, clustered codeword bits that would experience the same, or at least highly correlated fading are now “spread” to experience (ideally) i.i.d. fading, i.e., slow fading is in effect converted into fast fading to benefit decoding. The philosophy of BICM has been well applied in practice – for example, in the 3GPP UMTS and LTE systems with complicated wireless communication environments. Another widely-adopted way of using channel coding while balancing between error performance and bandwidth efficiency is the hybrid automatic repeat request (H-ARQ) process, in which, due to limited resources, different higher-rate redundancy versions of the original codeword are transmitted in a predefined order until the transmitter gets an acknowledgement (ACK) from the receiver. Here, channel coding not only protects the data, but also provides a mechanism for error detection in an early stage of receiving.

Compared to classical channel codes, some features of modern channel codes are as follows:

- Modern codes, such as Turbo codes and LDPC codes, are frequently referred to as “codes on graphs” [4]. As in the case of an LDPC code, a Tanner graph can be used to represent the parity-check structure of the code, which consists of variable nodes (representing the codeword bits/symbols), check nodes (representing the constraints that the values of the involved variable nodes must satisfy), and edges connecting a check node and the corresponding variable node(s). As for the Tanner graph of a Turbo code, refer to [4].

- Modern codes are praised for their near-capacity decoding performance; however, conventional optimal decoding algorithms for the codes, such as maximum-likelihood (ML) sequence decoding, are way too complicated for practical imple-
mentation, since the codeword length is typically very large. Instead, “bit/symbol-wise” iterative decoding algorithms are carried out on the code graphs: message is passed on an edge between the connected check node and variable node iteratively, representing the a-posteriori probability that the variable node takes a certain value; this “message-passing” fits into the general structure of belief propagation. In most cases, iterative decoding is suboptimal, and the code should avoid certain designs that have negative effects on the decoding such that the performance loss is acceptable, or even negligible. In fact, as channel coding develops from classical to modern, the “starting point” of the code design gradually shifted from encoding to decoding.

- Although greatly benefiting the practical implementation compared to ML or MAP (maximum a-posteriori) decoding, iterative decoding still contributes a large part of the total computational complexity in a communication system. This is critical especially when the data rate is high, and thus makes parallel decoding an implicit requirement. In the 3GPP standards, the UMTS system (known as the 3rd generation) and the LTE system (commercially known as the 4th generation) use Turbo codes with the same constituent convolutional encoders. However, “contention-free” interleavers based on quadratic polynomial permutation (QPP) are used in LTE so that parallel processors are able to access non-overlapping parts of the received codeword to achieve high throughput. As for LDPC codes, protograph-based design – one of the key techniques in this dissertation – is suitable for such cases. All of these, again, prove our previously-mentioned point that modern channel coding design focuses more on decoding.

This dissertation concentrates on modern channel codes – to be more specific, on LDPC codes. But it is worth pointing out that modern codes are not replacing classical codes:
• Modern and classical codes usually coexist in a sophisticated communication system and are applied to processing different types of data. Again, using LTE as an example, the Turbo code is used for the data channel while a tail-biting convolutional code is used for some of the control channels\(^1\); this is due to the fact that when the codeword length is small, the convolutional code shows a performance advantage, and thus is more suitable for control information since the system overhead is typically small.

• Modern and classical codes may have different error-correction capabilities, and thus can compensate each other when jointly processing the same dataflow. In optical communication, an LDPC code (the inner code) is often concatenated with a Reed-Solomon (RS) code (the outer code), so that during the jointly iterative decoding, the power of the RS code in the correction of burst errors can be exploited to improve the error-floor performance of the LDPC code.

• Modern and classical codes can even “merge” their structures to develop new types of codes. A perfect example for this is the focus of this dissertation, the spatially-coupled LDPC (SC-LDPC) code, also known as the terminated LDPC convolutional code. The name reveals the fact that it is a sparse graph code with memory, and we shall see that this structure results in capacity-approaching performance, even when there is no further optimization of the code according to the specific application.

1.2 Outline of the Dissertation

As previously mentioned in the abstract, this dissertation covers three topics around channel coding in wireless communication: 1) design of \(q\)-ary SC-LDPC codes for windowed decoding, 2) cooperative communication using binary LDPC block

\(^1\)Here “channel” refers to the data formatting rather than the environment for signal propagation.
codes with bandwidth-efficient modulation, and 3) the use of binary rate-compatible SC-LDPC codes in coded cooperation. These topics and all the related technical terms will be elaborated upon and discussed later in Chapters 2, 3, and 4, respectively. And in each of these chapters, we start with a chapter-wise introduction that describes the topic in more detail and states the problem that we are interested in, and then we present our design, show and analyze the theoretical and numerical results compared with conventional designs, and summarize the advantage of our design in the end. Finally, Chapter 5 summarize our contributions.

Throughout the dissertation,

- The wireless communication system operates in the baseband; signals and channel corruptions are discrete in time, but not necessarily in value – we do not consider fixed-point implementations in this dissertation but instead assume unquantized processing.

- For simplicity, we focus on the encoder/modulator in the transmitter and the demodulator/decoder in the receiver, with emphasis on the latter; “everything else” is perfect, i.e., signal distortion other than channel corruption, such as errors in synchronization, is beyond the scope of this dissertation.

As previously mentioned, this dissertation fits into the structure of “channel coding: design and application”:

- Chapter 2 focuses on code design for latency-constrained applications. Derived via theoretical analysis of windowed decoding, the design rules are simple and elegant to follow for constructions of practical finite-length codes.

- Chapter 3 focuses on code application in bandwidth-efficient cooperative communication. Inspired by BICM, we separate the encoder and the modulator (and thus separate the demodulator and the decoder), but unlike the conventional BICM with a pseudo-random interleaver, we use a multiplexer to combine
the local and relayed codewords in a predefined pattern (inspired by TCM) such that the demodulator is able to provide more reliable bit-wise initialization of the decoder.

- Chapter 4 returns to the binary coded cooperation system using binary LDPC codes, and proves that SC-LDPC codes are so powerful in general that capacity-approaching performance is achieved even if the code itself is not designed specially for the application.

To summarize, we would like to improve the performance of a wireless communication system using LDPC codes. Additional coding gain is provided either via an “internal” effort of making the coding scheme itself more powerful, or via an “external” effort of making the decoder essentially operate with higher signal-to-noise ratio.

We start with the design of $q$-ary SC-LDPC codes.
2.1 Introduction

Low-density parity-check block codes (LDPC-BCs) constructed over finite fields GF(q) of size $q > 2$ outperform comparable binary LDPC-BCs [5], in particular when the block length is short to moderate. However, this performance gain comes at the cost of an increase in decoding complexity. A direct implementation of the $q$-ary belief-propagation (BP) decoder, originally proposed by Davey and MacKay in [5], has complexity $O(q^2)$ per symbol. More recently, an implementation based on the fast Fourier transform [6] was shown to reduce the complexity to $O(q \log q)$. Beyond that, a variety of simple but sub-optimal decoding algorithms have been proposed in the literature, such as the extended min-sum (EMS) algorithm [7] and the trellis-based EMS algorithm [8]. For computing BP decoding thresholds, a $q$-ary extrinsic information transfer (EXIT) analysis was proposed in [9] and was later developed into a version suitable for protograph-based code ensembles [10].

A protograph [11] is a small Tanner graph, which can be used to produce a structured LDPC code ensemble by applying a graph lifting procedure [12] with lifting factor $M$, such that every code in the ensemble is $M$ times larger and maintains the structure of the protograph, i.e., it has the same degree distribution and the same type of edge connections. In this way, the computation graph [13] is maintained in the
Figure 2.1. A (3, 6)-regular protograph and its corresponding base-matrix representation. Black circles correspond to variable nodes and crossed boxes correspond to check nodes.

A protograph consisting of \((c - b)\) check nodes and \(c\) variable nodes has design rate \(R = b/c\) and can be represented equivalently by a \((c - b) \times c\) base (parity-check) matrix \(B\) consisting of non-negative integers, in which the \((i, j)\)-th entry \((1 \leq i \leq c - b\) and \(1 \leq j \leq c)\) is the number of edges connecting check node \(i\) and variable node \(j\).

Fig. 2.1 illustrates a (3, 6)-regular protograph and its corresponding base matrix, which can be used to represent a (3, 6)-regular LDPC-BC ensemble. To calculate the BP threshold of a protograph-based code ensemble, conventional tools are adapted to take the edge connections into account [11, 14]. Although some freedom is lost in the code design when the protograph structure is adopted, one can use these modified protograph-based analysis tools to find “good” protograph-based ensembles with better BP thresholds than corresponding unstructured ensembles with the same degree distribution [14, 15].

Spatially coupled LDPC (SC-LDPC) codes, also known as terminated LDPC convolutional codes [16], are constructed by coupling together a series of \(L\) disjoint, or uncoupled, LDPC-BC Tanner graphs. Binary SC-LDPC code ensembles have been
shown to exhibit a phenomenon called “threshold saturation” [17, 18, 19], in which, as the coupling length \( L \) grows, the BP decoding threshold saturates to the maximum \( a\text{-}posteriori \) (MAP) threshold of an uncoupled LDPC-BC ensemble, which in turn approaches the channel capacity as the density of the parity-check matrix increases [20]. This phenomenon has been reported for a variety of code ensembles (e.g., \((J,K)\)-regular code ensembles [21], accumulate-repeat-by-4-jagged-accumulate-based irregular code ensembles [22], bilayer code ensembles [23], and SC MacKay-Neal and SC Hsu-Anastasopoulos code ensembles [24]) and channel models (e.g., channels with memory [25], multiple access channels [26], intersymbol-interference channels [27], and the three-node erasure relay channel [28]), thus making SC-LDPC codes attractive candidates in practical applications requiring near-capacity performance. For a more comprehensive study of the literature, refer to the introduction part of [29]. In addition, more recently, Olmos et al. [30] used a scaling law to predict the finite-length performance of SC-LDPC codes on the binary erasure channel (BEC).

BP decoding threshold results on the BEC for \( q \)-ary SC-LDPC code ensembles have been reported by Uchikawa et al. [31] and Piemontese et al. [32], and the corresponding threshold saturation was proved by Andriyanova et al. [33]. In each of these papers, the authors assumed that decoding was simultaneously carried out across the entire parity-check matrix of the code; for simplicity, this will be referred to as flooding schedule decoding (FSD) in this chapter. Employing FSD for SC-LDPC codes can result in large latency, since large coupling length \( L \) is typically desired to achieve near-capacity thresholds [29]. To resolve this issue, a more efficient technique, called windowed decoding (WD), was proposed in [34, 35] for binary SC-LDPC codes. Compared to FSD, WD exploits the convolutional nature of the SC parity-check matrix to localize decoding and thereby reduce latency. Under WD, the decoding window contains only a small portion of the parity-check matrix, and within that window, BP decoding is performed.
In this chapter, assuming that the binary image of a codeword is transmitted, we analyze the WD thresholds of a variety of \((J,K)\)-regular protograph-based \(q\)-ary SC-LDPC code ensembles constructed from the corresponding uncoupled \(q\)-ary \((J,K)\)-regular LDPC-BC ensembles via the edge-spreading procedure [21, 29], where the finite field size is \(q = 2^m\) and \(m\) is a positive integer. In particular,

1. For the BEC, we first extend the \(q\)-ary density evolution (DE) analysis proposed in [36] to a protograph version, and apply this analysis in conjunction with a WD scheme to obtain \textit{windowed decoding thresholds} for \(q\)-ary SC-LDPC code ensembles;

2. For the binary-input additive white Gaussian noise channel (BIAWGNC) with binary phase-shift keying (BPSK) modulation, we obtain WD thresholds for \(q\)-ary SC-LDPC code ensembles by applying a protograph-based EXIT analysis (originally proposed for \(q\)-ary LDPC-BC ensembles [10]) in conjunction with a WD scheme.

In both cases, our primary contribution is to determine how much the decoding latency of WD can be reduced without suffering a loss in threshold. We observe that

1. Compared to FSD of the corresponding uncoupled \(q\)-ary LDPC-BC ensembles, WD of \(q\)-ary SC-LDPC code ensembles provides a threshold gain. This gain increases as the finite field size increases.

2. Compared to FSD of a given \(q\)-ary SC-LDPC code ensemble, WD provides significant reductions in both decoding latency and decoding complexity. Again, these reductions increase as the finite field size increases.

3. By carefully designing the protograph structure, using what we call the “type 2” edge-spreading format, WD provides near-capacity thresholds for \(q\)-ary SC-LDPC code ensembles, even when both the finite field size and the window size are relatively small.
4. When there is a constraint on decoding latency and operation close to the threshold of a binary SC-LDPC code ensemble is required, using the non-binary counterpart can provide a significant reduction in decoding complexity.

The rest of this chapter is organized as follows. Section 2.2 describes the construction of protograph-based $q$-ary SC-LDPC code ensembles and reviews WD. Then Sections 2.3 and 2.4 present the WD thresholds of various $q$-ary SC-LDPC code ensembles on the BEC and on the BIAWGNC, respectively, as the finite field size and/or the window size vary. The WD threshold is evaluated from two perspectives: first, as the window size increases, whether the WD threshold achieves its best numerical value when the window size is small to moderate; second, as the finite field size increases, whether this achievable value approaches capacity. Also, the effects of different protograph constructions on the WD thresholds are evaluated and discussed. Finally, Section 2.5 studies the decoding latency and complexity of $q$-ary SC-LDPC code ensembles and examines the tradeoffs of WD concerning latency, complexity, and performance.

In summary, by examining various $q$-ary SC-LDPC code ensembles, we bring additional insight to three questions:

1. Why spatially coupled codes perform better than the corresponding uncoupled block codes,

2. Why windowed decoding is preferred to flooding schedule decoding, and

3. When non-binary should be used instead of binary.

The results of this chapter provide theoretical guidance for designing and implementing practical $q$-ary spatially coupled LDPC codes suitable for windowed decoding [37].
2.2 Windowed Decoding of Protograph-Based $q$-ary SC-LDPC Code Ensembles

2.2.1 Protograph-based $q$-ary SC-LDPC Code Ensembles

A $(J, K)$-regular SC-LDPC code ensemble can be constructed from a $(J, K)$-regular LDPC-BC ensemble using the edge-spreading procedure [21, 29], described here first in terms of protograph representations of the code ensembles. Take $J = 3$, $K = 6$ as an example. As shown in Fig. 2.2, instead of transmitting a sequence of codewords from the $(3, 6)$-regular LDPC-BC ensemble independently at time instants $t = 1, 2, \ldots, L$, edges from the variable nodes at time instant $t$, originally connected only to the check node at time instant $t$, are now “spread” to also connect to check nodes at time instants $t, t+1, \ldots, t+w$; in this way, memory is introduced and the different time instants are “coupled” together, i.e., a terminated convolutional, or spatially coupled, coding structure is introduced. The parameter $w$ is referred to as the coupling width, and $L$ is called the coupling length. Fig. 2.2 shows three different types of edge-spreading formats for $w = 1$ and one type for $w = 2$, all for the case $J = 3$, $K = 6$, and $L = 8$.

The above edge-spreading procedure can be described in terms of the base (parity-check) matrix representation of protographs as well. Let $B$ be a $(c-b) \times c$ block base matrix representing a LDPC-BC ensemble with design rate $R = b/c$. Then the base matrix of an SC-LDPC code ensemble can be constructed from $B$ as follows. First, $B$ is “spread” into a set of $(w + 1)$ component base matrices following the rule

$$\sum_{i=0}^{w} B_i = B,$$  \hspace{1cm} (2.1)

so that each $B_i$ has the same size as $B$. Next, an SC base matrix $B_{SC}$ is generated by “stacking and shifting” the base component matrices $\{B_i\}_{i=0}^{w}$ at each time instant.
Figure 2.2. (a) A sequence of $L = 8$ uncoupled $(3, 6)$-regular LDPC-BC protographs, and (b)-(e) various $(3, 6)$-regular SC-LDPC protographs constructed following the edge-spreading procedure with the coupling length $L = 8$. 

- (a) $w=2$: $B_0 = B_1 = B_2 = [1 \ 1]$ 
- (b) $w=1$, type 1: $B_0 = [1 \ 1], B_1 = [2 \ 2]$ 
- (c) $w=1$, type 2: $B_0 = [1 \ 2], B_1 = [2 \ 1]$ 
- (d) $w=1$, type 3: $B_0 = [2 \ 2], B_1 = [1 \ 1]$ 
- (e) $w=1$, type 4: $B_0 = [2 \ 1], B_1 = [1 \ 2]$
$t = 1, 2, \ldots, L$, thereby forming a convolutional structure:

$$\mathbf{B}_{SC} = \begin{bmatrix}
\mathbf{B}_0 \\
\mathbf{B}_1 & \mathbf{B}_0 \\
\vdots & \mathbf{B}_1 & \ddots \\
\mathbf{B}_w & \mathbf{B}_1 & \cdots & \mathbf{B}_0 \\
\mathbf{B}_w & \mathbf{B}_1 & \cdots & \mathbf{B}_w
\end{bmatrix}_{(L+w)(c-b) \times Lc}, \quad (2.2)$$

where the design rate of $\mathbf{B}_{SC}$ is

$$R_L = 1 - \frac{(L + w)(c - b)}{Lc} = \frac{Lb - w(c - b)}{Lc}. \quad (2.3)$$

Due to the termination of $\mathbf{B}_{SC}$ after $Lc$ columns, there exists a loss in the SC-LDPC code ensemble design rate $R_L$ compared to the rate $R$ of $\mathbf{B}$. However, this rate loss diminishes as $L$ increases and vanishes as $L \to \infty$, i.e., $\lim_{L \to \infty} R_L = R = b/c$.

Then a finite-length $q$-ary SC-LDPC code is constructed from $\mathbf{B}_{SC} = [b_{i,j}]$ by following the procedure for constructing a finite-length $q$-ary LDPC-BC from $\mathbf{B}$:

1. “Lifting” [11]: Replace the nonzero entries $b_{i,j}$ in $\mathbf{B}_{SC}$ with an $M \times M$ permutation matrix (or a sum of $b_{i,j}$ non-overlapping $M \times M$ permutation matrices if $b_{i,j} > 1$), and replace the zero entries by the $M \times M$ all-zero matrix, where $M$ is called the lifting factor.

2. “Labeling”: Randomly assign to each non-zero entry in the lifted parity-check matrix a non-zero element from $\text{GF}(q)$, where $q = 2^m$ is the finite field size.

After the lifting step, the parity-check matrix is still binary, i.e., the non-binary
Figure 2.3. WD example with window size $W = 3$: at $t = 1$ (solid red), $t = 2$ (dotted blue), and $t = 3$ (dash green). $J = 3$, $K = 6$, $w = 1$; $B_0 = [1, 1]$ and $B_1 = [2, 2]$, both of size $(c - b) \times c = 1 \times 2$, for construction of $B_{SC}$ (2.2). For each window position/time instant, the first $c = 2$ column blocks are target symbols.

feature does not arise until the labeling step.\footnote{1} The total code length is $n = LcM$, and we define the constraint length as the maximum width of the non-zero portion of the parity-check matrix $\nu = (w + 1)cM$. Both the permutation matrices and the $q$-ary labels can be carefully chosen to obtain good codes with desirable properties. But this is not our emphasis; rather, we are interested in a threshold analysis of general $q$-ary ensembles consisting of all possible combinations of liftings and labelings for a protograph, and the dimension of the message model used in the analysis depends on the size of the finite field [9, 36].
2.2.2 Windowed Decoding (WD)

In this subsection, we briefly review the structure of WD.

By construction, any two variable nodes (columns of the parity-check matrix) in the graph of an SC-LDPC code cannot be connected to the same check if they are more than a constraint length \( \nu = (w+1)cM \) (of columns) apart. As previously mentioned, compared to flooding-schedule decoding (FSD) where the iterative decoding is carried out on the entire parity-check matrix, WD of SC-LDPC code ensembles takes advantage of the convolutional structure of the parity-check matrix and localizes the decoding process to a small portion of the matrix, i.e., the BP algorithm is carried out only for those checks and variables covered by a “window”. Consequently, WD is an efficient way to reduce the memory and latency requirements for implementations of SC-LDPC codes [34, 35]. The WD algorithm can be described as follows (see [34] for further details):

- Described using the SC base matrix \( \mathbf{B}_{\text{SC}} \), the window is of fixed size \((c-b)W \times (c-b)W\) (recall that the size of the component base matrices \( \mathbf{B}_i \)'s in \( \mathbf{B}_{\text{SC}} \) is \((c-b) \times c\)) measured in symbols, and slides from time instant \( t = 1 \) to time instant \( t = L \), where \( W \), called the window size, is defined as the number of column blocks of size \( c \) in the window. An example of WD with \( W = 3 \) is illustrated in Fig. 2.3, for the SC-LDPC code ensemble with protograph shown as (c) in Figure 2.2.

- At each time instant/window position, the BP algorithm runs until a fixed number of iterations has been performed or some stopping rule [34, 35, 37] is satisfied, after which the window shifts \( c \) column blocks and those \( c \) column block symbols shifted out of the window are decoded. The first \( c \) column blocks

\[ ^1 \text{Note that “labeling” can come before “lifting”, resulting in a “constrained” protograph-based } q\text{-ary code as defined in [10].} \]
in any window are called *target symbols*. We assume that all the variables and checks in a window are updated during each iteration, and messages are passed to the window from previously decoded symbols where applicable.

- Clearly, the largest possible $W$ is equal to $(L + w)$, where the whole parity check matrix is covered and makes WD equivalent to FSD, and the smallest possible $W$ is $(w + 1)$, i.e., the window length (measured in variables) when decoding an SC-LDPC code must be at least on constraint length. We are interested in searching for $q$-ary SC-LDPC code ensembles for which a small $W$ can provide WD with a good threshold, which suggests that $w$ should be kept small. Indeed, our results for $q$-ary SC-LDPC codes together with those in the literature for binary SC-LDPC codes [34, 35] show that ensembles with $w = 1$ provide the best latency-constrained performance with WD.

2.2.3 Code Ensemble Construction

In this chapter, we restrict our attention to $(J,K)$-regular LDPC code ensembles.

2.2.3.1 $(J,K)$-Regular LDPC-BC Ensembles

Let

$$B = \begin{bmatrix} J & J & \cdots & J \end{bmatrix}_{1 \times k}$$  \hspace{1cm} (2.4)

denote the block base matrix corresponding to the protograph representation of a $(J,K)$-regular LDPC-BC ensemble, where $K = kJ$, $k = 1, 2, \ldots$, and the design rate of the code ensemble is $R = (k - 1)/k$. That is, in the remainder of this chapter, let $c - b = 1$ and $c = k$. We denote the $(J,K)$-regular LDPC-BC ensemble constructed over $GF(2^m)$ as $B(J,K,m)$. 

18
2.2.3.2 Edge Spreadings of $B$

Given a variable node degree $J$, for a particular coupling width $w$, define

$$E(J, w) = \left\{ \left[ \begin{array}{cccc} J_0 & J_1 & \cdots & J_w \end{array} \right]^\top \mid \sum_{i=0}^{w} J_i = J, J_i \in \{1, 2, \ldots, J - w\} \right\},$$

(2.5)

i.e., $E(J, w)$ is the set of all possible column vectors of length $(w + 1)$ satisfying the constraint $\sum_{i=0}^{w} J_i = J$, where $J_i \in \{1, 2, \ldots, J - w\}$. Moreover, define $B^w_0$ as

$$B^w_0 = \begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_w \end{bmatrix}_{(w+1) \times k},$$

(2.6)

i.e., $B^w_0$ is the “stack” of all the component base matrices $\{B_i\}_{i=0}^{w}$. Then an edge-spreading format can be generated by selecting $k$ elements (with replacement) from $E(J, w)$ as the columns of $B^w_0$. Recall that our major interest lies in $q$-ary SC-LDPC code ensembles for which windowed decoding (WD) achieves good thresholds under tight latency constraints, i.e., with a small window size $W$. In this case, as mentioned in Section 2.2.2, it is already well known that the threshold can be improved by using small $w$. Therefore, we do not allow $w$ to exceed $(J - 1)$, i.e., the block base matrix $B$ should be spread into at most $J$ component base matrices $B_i$. In other words, for $E(J, w)$ in (2.5), let $1 \leq w \leq J - 1$.

The edge-spreading format $B^w_0$ determines the SC base matrix $B_{SC}$, and the $q$-ary density evolution algorithm depends on $B_{SC}$. For a given $B^w_0$, its column permutation does not affect the WD threshold, but its row permutation does. Consequently, for each combination of $J$ and $w$, to result in different WD thresholds, there will be $|E(J, w)| \cdot (1 + |E(J, w)|) / 2$ possible edge-spreading formats. For example, consider
the \((4, 8)\)-regular degree distribution with \(J = 4\) and \(w = 2\). Then

\[
E(4, 2) = \left\{ \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}^\top, \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}^\top \right\},
\]

(2.7)

and the \(|E(4, 2)| \cdot (1 + |E(4, 2)|) / 2 = 6\) possible edge-spreading formats are given by

\[
B_0^w \in \left\{ \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 2 & 2 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \right\},
\]

(2.8)

with different WD thresholds.

2.2.3.3 \((J, K)\)-Regular SC-LDPC Code Ensembles

In this part, we detail the particular constructions of SC-LDPC code ensembles considered in the remainder of this chapter. The first construction we consider is the “classical” edge spreading [17] of the \((J, K)\)-regular LDPC-BC base matrix (2.4), where \(K = kJ\) and \(w = J - 1\):

\[
B_0 = B_1 = \cdots = B_w = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}_{1 \times k}.
\]

(2.9)

Unless noted otherwise, the coupling length for all the \(q\)-ary SC-LDPC code ensembles in this chapter is taken to be \(L = 100\), in order to keep the rate loss small. Consequently, we do not include \(L\) in the ensemble notation, and we denote the SC-LDPC code ensemble constructed over \(GF(2^m)\) using component matrices (2.9) in (2.2), with coupling width \(w = J - 1\), as \(C_{J-1}(J, K, m)\).

Again, as previously mentioned, under tight latency constraints, the WD threshold can be improved by using small \(w\); in fact, excellent WD performance has been shown for binary SC-LDPC code ensembles using repeated edges in the protograph.
and $w = 1$ [34, 35]. Consequently, we focus on the case $w = 1$ for the remaining constructions, i.e., we consider
\[
E(J, w = 1) = \left\{ \begin{bmatrix} 1 \\ J-1 \end{bmatrix}, \begin{bmatrix} 2 \\ J-2 \end{bmatrix}, \ldots, \begin{bmatrix} J-1 \\ 1 \end{bmatrix} \right\}.
\] (2.10)

For clarity, we further restrict our attention to the edge-spreading pair
\[
E_A = \begin{bmatrix} 1 \\ J-1 \end{bmatrix}, \quad E_B = \begin{bmatrix} J-1 \\ 1 \end{bmatrix} \in E(J, 1).
\] (2.11)

We have found in our studies that combinations of $E_A$ and $E_B$ result in the most interesting and representative constructions compared to general selections of column vectors from $E(J, 1)$.

Combining $E_A$ and $E_B$, there are $(k+1)$ possible choices for $B_0^w=1$. An edge-spreading format is called “type-$p$” if there are $(k-p+1)$ columns of $E_A$ in $B_0^1$ followed by $(p-1)$ columns of $E_B$, i.e.,
\[
B_0^1 = \begin{bmatrix} E_A & \cdots & E_A & E_B & \cdots & E_B \end{bmatrix}_{k-p+1,p-1} = \begin{bmatrix} 1 & \cdots & 1 & J-1 & \cdots & J-1 \\ J-1 & \cdots & J-1 & 1 & \cdots & 1 \end{bmatrix}_{k-p+1,p-1} = \begin{bmatrix} B_0 \\ B_1 \end{bmatrix},
\] (2.12)

where $1 \leq p \leq k+1$. Again, note that the ordering of columns is not important, because this simply results in column permutations of the resulting base matrix (2.2) and does not change code or graph properties. We omit $L$ from the ensemble notation, and denote the type-$p$ SC-LDPC code ensemble constructed over $\text{GF}(2^m)$ using component matrices $B_0$ and $B_1$ in (2.2), with coupling width $w = 1$, as $C_1(J, K, m, p)$,
where $1 \leq p \leq k + 1$.

For a particular $(J, K)$ pair and Galois field $GF(2^m)$, we refer informally to the collection of ensembles

$\{ \mathcal{B}(J, K, m), \mathcal{C}_{J-1}(J, K, m), \mathcal{C}_1(J, K, m, p) \mid p = 1, 2, ..., k + 1 \}$ \hspace{1cm} (2.13)

as “the $(J, K, m)$ ensembles”, and refer to the collection of ensembles

$\{ \mathcal{C}_{J-1}(J, K, m), \mathcal{C}_1(J, K, m, p) \mid p = 1, 2, ..., k + 1 \}$ \hspace{1cm} (2.14)

as “the $(J, K, m)$ SC ensembles”. For example, for an arbitrary $m$, let $(J, K) = (3, 6)$. In this case $k = 2$, and, in addition to the classical edge spreading with $w = J - 1 = 2$, there are $k + 1 = 3$ types of edge-spreading for $w = 1$:

- $\mathcal{C}_2(3, 6, m)$: $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$;
- $\mathcal{C}_1(3, 6, m, 1)$: $\mathbf{B}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $\mathbf{B}_1 = \begin{bmatrix} 2 & 2 \end{bmatrix}$;
- $\mathcal{C}_1(3, 6, m, 2)$: $\mathbf{B}_0 = \begin{bmatrix} 1 & 2 \end{bmatrix}$, $\mathbf{B}_1 = \begin{bmatrix} 2 & 1 \end{bmatrix}$;
- $\mathcal{C}_1(3, 6, m, 3)$: $\mathbf{B}_0 = \begin{bmatrix} 2 & 2 \end{bmatrix}$, $\mathbf{B}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

These ensembles form the $(3, 6, m)$ SC ensembles, and together with $\mathcal{B}(3, 6, m)$, form the $(3, 6, m)$ ensembles. Fig. 2.2 illustrates the $(3, 6, m)$ ensembles with coupling length $L = 8$ and an arbitrary $m$.

2.3 Threshold Analysis of $q$-ary SC-LDPC Code Ensembles on the BEC

2.3.1 Protograph $q$-ary Density Evolution (DE) on the BEC

The $q$-ary DE algorithm presented in [36] was originally derived for unstructured/randomized uncoupled $q$-ary LDPC-BC ensembles where 1) the symbol set is
the vector space $\text{GF}_2^m$ of dimension $m$ over the binary field, and 2) the edge labeling set is the general linear group $\text{GL}_2^m$ over the binary field, which is the set of all $m \times m$ invertible matrices whose entries are in $\{0, 1\}$. The thresholds of these code ensembles, as pointed out by the authors of [36], are very good approximations to those of $q$-ary LDPC-BC ensembles defined over $\text{GF}(2^m)$, since the numerical difference is on the order of $10^{-4}$. Thus, we modify this $q$-ary DE algorithm to calculate the thresholds of various protograph-based $(J, K)$-regular $q$-ary SC-LDPC code ensembles defined over $\text{GF}(2^m)$ in this chapter.

Consider an ordered list of the elements of $\text{GF}_2^m$, and assume that the zero element is in the zeroth position of the list. For a finite length code, a probability domain message vector in $q$-ary BP decoding is of length $2^m$, where the entry at position $i$ corresponds to the a posteriori probability that the symbol is the $i$-th element from $\text{GF}_2^m$. Since it can be assumed that the all-zero codeword is transmitted without affecting the decoding performance [36] and transmission is on the BEC, all the non-zero elements in the message vector must be equal; in fact, the set of those symbols (elements from $\text{GF}_2^m$) whose a posteriori probabilities are non-zero forms a subspace of $\text{GF}_2^m$. The message vector is then said to have dimension $n$ if it contains $2^n$ non-zero elements, $n = 0, 1, \ldots, m$. Consequently, for the purpose of $q$-ary DE which studies the ensemble-average properties rather than a specific finite-length code, only the dimension of the BP decoding message vector needs to be tracked in the algorithm. As a result, a $q$-ary DE message vector for the BEC can be represented by a vector of length $(m+1)$, whose $j$-th entry, $j = 0, 1, \ldots, m$, indicates the a posteriori probability that the BP decoding message vector has dimension $n$.

2.3.1.1 Protograph DE for $q$-ary LDPC Code Ensembles

Similar to the procedure used to extend $q$-ary EXIT analysis to a protograph version in [10], in this chapter, we extend the $q$-ary DE algorithm to a protograph
version, which we refer to as $q$-ary protograph DE (PDE). Since the edge connections are taken into account and the computation graph is equal for all code members of the ensemble, PDE reduces to the BP algorithm performed on the protograph. We use notation similar to that in [10] and [33]. Let $b_{i,j}$ denote a non-zero entry in the (portion of the) base matrix where the algorithm is carried out; recall that from the perspective of the protograph, the value of $b_{i,j}$ is the number of edges connecting check node $i$ (the row index in the matrix) to variable node $j$ (the column index), rather than an edge label. Let $N(i)$ (resp. $M(j)$) denote the neighboring variables (resp. checks) of check $i$ (resp. variable $j$). Let $p_C^{(l)}(i,j)$ (resp. $p_V^{(l)}(i,j)$) denote the check-i-to-variable-j (resp. variable-j-to-check-i) $q$-ary DE message vector during iteration $l$. Finally, let the erasure probability of the BEC be $\epsilon$. Then the $q$-ary PDE algorithm consists of four steps as follows:

- **Initialization:** for each $b_{i,j} > 0$, let

\begin{align}
\hat{p}_V^{(0)}(i,j) &= p_C^{(0)}(i,j) = p_V^{(0)}(\epsilon), 
\end{align}

where $p^{(0)}(x)$ is a vector of length $(m + 1)$ in the probability domain, whose $n$-th entry is defined as

\begin{align}
\binom{m}{n} x^n(1-x)^{m-n}.
\end{align}

- **Check-to-variable update:** the message vector from check $i$ to variable $j$ is

\begin{align}
\hat{p}_C^{(l)}(i,j) &= \left[ \bigotimes_{s \in N(i) \setminus j} \left( \bigotimes_{s \in N(i) \setminus j} p_V^{(l-1)}(i,s) \right) \right] \bigotimes \left( \bigotimes_{s \in N(i) \setminus j} p_V^{(l-1)}(i,s) \right),
\end{align}

where the “⊗” notation (see Appendix A of [33] for details) is described as
follows. For two $q$-ary DE message vectors $\mathbf{p}_1$ and $\mathbf{p}_2$, $\mathbf{p}_1 \boxtimes \mathbf{p}_2$ has $n$-th element

$$\sum_{i=0}^{n} \sum_{j=n-i}^{n} C_{i,j,n}^m p_{1,i} p_{2,j}, \quad (2.18)$$

where $p_{1,i}$ is the $i$-th element of $\mathbf{p}_1$, $p_{2,j}$ is the $j$-th element of $\mathbf{p}_2$,

$$C_{i,j,n}^m = \frac{G_{m-i,m-n} G_{i,n-j} 2^{(n-i)(n-j)}}{G_{m,m-j}} \quad (2.19)$$

is the probability of choosing a subspace (of $\text{GF}_2^m$) of dimension $j$ whose sum with a subspace of dimension $i$ has dimension $n$, and

$$G_{m,k} = \begin{cases} 1 & \text{if } k = m \text{ or } k = 0, \\ \prod_{i=0}^{k-1} \frac{2^m - 2^i}{2^k - 2^i} & \text{if } 0 < k < m, \\ 0 & \text{otherwise}, \end{cases} \quad (2.20)$$

is the Gaussian binomial coefficient, the number of different subspaces of dimension $k$ of $\text{GF}_2^m$. Finally, $\boxtimes^{b_{i,j}-1} \mathbf{p} = \mathbf{p} \boxtimes \mathbf{p} \boxtimes \ldots \boxtimes \mathbf{p}$, with $(b_{i,j} - 1)$ occurrences of $\mathbf{p}$.

- Variable-to-check update: the message vector from variable $j$ to check $i$ is

$$\mathbf{p}_V^{(l)} (i,j) = \mathbf{p}_V^{(0)} (j) \square \left[ \Box_{s \in M(j) \setminus i} \left( \Box_{s \leq j} \mathbf{p}_C^{(l)} (s,j) \right) \right] \Box \left( \Box_{s \leq j} \mathbf{p}_C^{(l)} (s,j) \right), \quad (2.21)$$

where $\mathbf{p}_1 \square \mathbf{p}_2$ has $n$-th element

$$\sum_{i=n}^{m} \sum_{j=n}^{m-i+n} V_{i,j,n}^m p_{1,i} p_{2,j}, \quad (2.22)$$
and

\[ V_{i,j,n}^m = \frac{G_{i,n}G_{m-i,j-n}2^{(i-n)(j-n)}}{G_{m,j}} \]  

(2.23)

is the probability of choosing a subspace of dimension \( j \) whose intersection with a subspace of dimension \( i \) has dimension \( n \) (again, see Appendix A of [33] for details).

- Convergence check: the a-posteriori message vector for variable \( j \) is

\[ p_{V,\text{APP}}^{(l)}(j) = p_{V}^{(0)}(j) \otimes \left[ \otimes_{i\in M(j)} \left( \otimes_{b_{i,j}} p_{C}^{(l)}(i,j) \right) \right]. \]  

(2.24)

The \( q \)-ary PDE algorithm ends when

- Either a decoding success is declared: for all the variables to be decoded, the 0th entry of each \( p_{V,\text{APP}}^{(l)}(j) \) (denoted as \( p_{V,\text{APP}}^{(l)}(j)[0] \)) is at least \((1-\delta)\), i.e., \( p_{V,\text{APP}}^{(l)}(j)[0] \geq 1-\delta \), where \( \delta \in [0,1] \) is a preset erasure rate,

- Or a decoding failure is declared: the algorithm reaches some maximum number of iterations.

The parameter \( \delta \) should be chosen small enough so that it is essentially certain that \( q \)-ary PDE has converged if the condition is satisfied.

2.3.1.2 Flooding-Schedule Decoding (FSD) Thresholds for \( q \)-ary SC-LDPC Code Ensembles

Given \( m \) characterizing the symbol set and \( \epsilon \) characterizing the BEC, if \( q \)-ary PDE is operated on the entire \( B_{\text{SC}} \) for an SC-LDPC code ensemble, then the algorithm determines asymptotically (i.e., the coupling length \( L \rightarrow \infty \) and the lifting factor \( M \rightarrow \infty \)), whether FSD can be successful on the ensemble average for that specific
BECE. Thus, $q$-ary PDE can be used to calculate the FSD threshold, denoted $\epsilon^*(m, \delta)$, which is the largest channel erasure rate such that all transmitted symbols can be recovered successfully with probability at least $(1 - \delta)$, as the number of iterations $l$ goes to infinity:

$$\epsilon^*(m, \delta) = \sup \left\{ \epsilon \in [0, 1] \mid p^{(l)}_{V, \text{APP}}(j)[0] \geq 1 - \delta \text{ for } 1 \leq j \leq kL, \text{ as } l \to \infty \right\} . \tag{2.25}$$

The following numerical threshold results on the BEC are obtained for $\delta = 10^{-6}$, and from this point forward, $\epsilon^*(m, \delta)$ will be denoted simply as $\epsilon^*(m)$.

### 2.3.1.3 Windowed Decoding (WD) Thresholds for $q$-ary SC-LDPC Code Ensembles

We also incorporate $q$-ary PDE into WD and generate a $q$-ary WD-PDE algorithm, in order to calculate the WD threshold of an SC-LDPC code ensemble defined over $\text{GF}(2^m)$.

The $q$-ary WD-PDE algorithm consists of operation of $q$-ary PDE for all the window positions/time instants $t = 1, 2, \ldots, L$, as illustrated in Fig. 2.3. For each window position, $q$-ary PDE runs within the $W \times kW$ window; however, unlike the case of FSD, now the convergence check involves only the target symbols, i.e., the first $k$ symbols in the window. Starting from $t = 1$, if $q$-ary PDE declares a decoding failure, then the whole $q$-ary WD-PDE terminates and declares a decoding failure; otherwise, the window slides forward and $q$-ary PDE runs for the next window position. $q$-ary WD-PDE declares a decoding success for a specific BEC with $\epsilon$ if and only if its “component” $q$-ary PDE declares decoding successes for all the window positions. Thus, given $m, \epsilon, \text{ and } W$, $q$-ary WD-PDE can be used to calculate the WD threshold of an SC-LDPC code ensemble.
To be more specific, define

$$
\epsilon_{WD}^*(m, W, t, \delta) = \sup \left\{ \epsilon \in [0, 1] \left| p_{V, APP}^{(l)}(j)[0] \geq 1 - \delta \text{ for } tk - k + 1 \leq j \leq tk, \text{ as } l \to \infty \right. \right\}
$$

(2.26)

as the largest channel erasure rate such that all the target symbols in window position $t$ can be recovered successfully with probability at least $(1 - \delta)$, as $l$ goes to infinity, given that all the target symbols in the previous $(t - 1)$ windows have already been recovered successfully with probability at least $(1 - \delta)$. Then the WD threshold

$$
\epsilon_{WD}^*(m, W, \delta) = \inf_{1 \leq t \leq L} \epsilon_{WD}^*(m, W, t, \delta),
$$

(2.27)

guaranteeing that all the transmitted symbols – consisting of all the target symbols of all the windows – can be recovered successfully with probability at least $(1 - \delta)$, as $l$ goes to infinity.

Similar to the case of FSD threshold, $\delta = 10^{-6}$, and from this point forward, $\epsilon_{WD}^*(m, W, \delta)$ will be denoted simply as $\epsilon_{WD}^*(m, W)$.

2.3.2 Numerical Results: $k = 2$ ($R = 1/2$)

In this subsection we focus on the BP thresholds of the rate $R = 1/2$ q-ary SC-LDPC code ensembles with $k = 2$: in particular, we consider the $(2, 4)$-, $(3, 6)$-, $(4, 8)$-, and $(5, 10)$-regular code ensembles. Our emphasis is on the scenario when WD is used, and the $q$-ary WD-PDE algorithm described in the previous subsection is adopted to calculate the corresponding BP threshold.

Recall from Section 2.2.3 that, for $k = 2$, the SC-LDPC code ensembles we
The classical spreading results in the maximum coupling width \( w \) by making all the elements in \( B_0^w \) equal to 1. As for \( w = 1 \), clearly, the type 1 and type 3 ensembles, \( \mathcal{C}_1(J, K, m, 1) \) and \( \mathcal{C}_1(J, K, m, 3) \), will have the same FSD threshold \( \epsilon^*(m) \), since their SC base matrices are equal up to row permutations and the \( q \)-ary PDE algorithm runs over the entire \( B_{SC}^w \). However, their WD thresholds are different. Type 2 has one column of \( B_0^w \) that is the same as type 1 and the other column that is the same as type 3, so it is expected that its WD threshold will be between those of types 1 and 3.

2.3.2.1 The \((2, 4)\) Ensembles

When \((J, K) = (2, 4)\), all four types of edge spreading (2.28)-(2.31) for \( q \)-ary SC-LDPC code ensembles are the same. For \( m = 1, 2, \ldots, 10 \), the FSD and WD thresholds are shown in Fig. 2.4:

- Comparing \( \mathcal{C}_1(2, 4, m) \) to \( B(2, 4, m) \), the improvement in the FSD threshold \( \epsilon^*(m) \) introduced by the spatially coupled structure is negligible for small \( m \). However, as \( m \) increases, \( \epsilon^*(m) \) of \( \mathcal{C}_1(2, 4, m) \) increases and approaches the BEC capacity for a rate \( R = 1/2 \) code ensemble.\(^2\) This is consistent with the

\[^2\]Since \( L = 100 \), the design rate of \( \mathcal{C}_1(2, 4, m) \) is \( R_L = 0.495 \) and capacity is \( \epsilon_{Sh} = 1 - 0.495 = 0.505 \). This gap to capacity vanishes as \( L \to \infty \), since the thresholds do not further decay and \( R_L \to 1/2 \).
observations made in [31]. We note that the $\mathcal{B}(2, 4, m)$ ensembles do not display this behavior as their thresholds diverge from capacity as $m$ increases, $m \geq 5$.

- For WD of $\mathcal{C}_1(2, 4, m)$ with fixed $m$, the threshold $\epsilon_{\text{WD}}^*(m, W)$ improves as the window size $W$ increases and saturates numerically to a (maximum) constant value $\hat{\epsilon}_{\text{WD}}(m)$. Thus, we define

$$W^*(m) = \min \{W \mid \epsilon_{\text{WD}}^*(m, W) \cong \hat{\epsilon}_{\text{WD}}(m)\}$$

(2.32)

as the smallest window size that provides the best threshold $\hat{\epsilon}_{\text{WD}}(m)$ for a fixed $m$; here, “$\cong$” is used for a numerically indistinguishable equality.\(^3\) We now make three observations regarding the ensemble $\mathcal{C}_1(2, 4, m)$:

- For all $m$, $\hat{\epsilon}_{\text{WD}}(m) = \epsilon^*(m)$, i.e., when the window size $W$ is large enough,

---

\(^3\)For our purposes, two thresholds are numerically indistinguishable if their absolute difference is no more than $10^{-6}$.  

---
the WD threshold equals the FSD threshold.

– As $m$ increases, $W^*(m)$ is non-increasing, i.e., increasing the finite field size can “speed up” the saturation of $\hat{\epsilon}_{\text{WD}}(m,W)$ to $\hat{\epsilon}_{\text{WD}}(m)$ as $W$ increases.

– The saturation of $\hat{\epsilon}_{\text{WD}}(m,W)$ to $\hat{\epsilon}_{\text{WD}}(m)$ is relatively slow as $W$ increases, especially when $m$ is small. For example, when $m = 1$, we need a window size of $W^*(1) = 30$ to obtain the threshold $\hat{\epsilon}_{\text{WD}}(1)$. Moreover, even for a fairly large window, say $W = 20$, the WD threshold of $C_1(2,4,m)$ is worse than the FSD threshold of $B(2,4,m)$ for $m = 1, 2, \text{and} 3$. This indicates that $C_1(2,4,m)$ does not perform well unless $W$ and/or $m$ are large.

As a result, we conclude that $C_1(2,4,m)$ is not a good candidate for WD, since a desirable $q$-ary SC-LDPC code ensemble should provide a near-capacity threshold when both the finite field size and the window size are relatively small, resulting in both small decoding latency and small decoding complexity – details will be discussed later in Section 2.5. We will see in the remainder of this section that increasing the node degrees in the code graph speeds up the saturation of $\hat{\epsilon}_{\text{WD}}(m,W)$ to $\hat{\epsilon}_{\text{WD}}(m)$.

2.3.2.2 The (3,6) Ensembles

As a benchmark, Fig. 2.5 compares the FSD thresholds of ensembles $C_1(3,6,m,1)$ (and thus $C_1(3,6,m,3)$) and $C_1(3,6,m,2)$ to that of $B(3,6,m)$ for various $m$.\(^4\) It is observed that:

- For all finite field sizes $2^m$, the introduction of the spatially coupled structure provides all four $q$-ary SC-LDPC code ensembles with significant improvement in the FSD threshold compared to the $q$-ary LDPC-BC ensemble. In fact, the

\(^4\)Code ensembles $C_2(3,6,m)$ are not included in Fig. 2.5, because their thresholds are almost indistinguishable (although slightly different) from those of $C_1(3,6,m,1)$.
gap between $\mathcal{B}(3, 6, m)$ and the $(3, 6, m)$ SC ensembles increases as $m$ increases. Again, this is consistent with the observations made in [31] and [32].

- Note that, like the $\mathcal{B}(2, 4, m)$ ensembles discussed above, the $\mathcal{B}(3, 6, m)$ thresholds diverge from capacity as $m$ increases. The FSD thresholds of $\mathcal{C}_1(3, 6, m, 1)$ and $\mathcal{C}_2(3, 6, m)$ increase and approach the BEC capacity for rate $R = 1/2$ as $m$ increases. Surprisingly, this is not the case for $\mathcal{C}_1(3, 6, m, 2)$, whose FSD threshold increases and approaches capacity for $m = 5$, but then decreases slowly and thus diverges slightly from capacity as $m$ increases further. As a result, in Fig. 2.5, there exists a small gap between the thresholds of $\mathcal{C}_1(3, 6, m, 1)$ and $\mathcal{C}_1(3, 6, m, 2)$ for large $m$.

We continue our study by briefly demonstrating the FSD threshold behavior of SC-LDPC code ensembles for varying coupling lengths $L$. Fig. 2.6 shows the FSD thresholds $\epsilon^*(m)$ for ensembles $\mathcal{C}_1(3, 6, m, 2)$ and $\mathcal{C}_2(3, 6, m)$ with increasing $L$. For fixed $m$ and increasing $L$, the FSD thresholds initially decrease and then saturate to
Figure 2.6. FSD thresholds of SC-LDPC code ensembles with different coupling lengths $L$. 

(a) $\mathcal{C}_1(3,6,m,2)$. 
(b) $\mathcal{C}_2(3,6,m)$. 

33
a constant value for sufficiently large $L$. This is consistent with results for binary protograph-based SC-LDPC code ensembles [17, 29] and, for closely related “randomized” SC-LDPC code ensembles, the saturation value is known to be the MAP decoding threshold for the uncoupled LDPC-BC ensemble for both binary [18] and non-binary [33] cases.

Regarding design of the edge-spreading format, this threshold saturation clearly illustrates the point made above that the $C_1(3, 6, m, 2)$ ensemble does not have monotonically increasing thresholds with $m$. In Fig. 2.6(a), for $C_1(3, 6, m, 2)$, we have $\epsilon^*(1) < \epsilon^*(5)$ but $\epsilon^*(10) < \epsilon^*(5)$, while in Fig. 2.6(b), for $C_2(3, 6, m)$, $\epsilon^*(m)$ increases uniformly as $m$ increases: this confirms our observation of the small gap between the FSD thresholds of $C_1(3, 6, m, 1)$ (almost indistinguishable from $C_2(3, 6, m)$) and $C_1(3, 6, m, 2)$ for large $m$ noted in Fig. 2.5.

Given $m$, let $L^*(m)$ be the minimum $L$ such that the threshold has saturated to its constant value, i.e.,

$$L^*(m) = \min \{ L \mid \epsilon^*(m, L) \cong \epsilon^*(m, L'), L' = L + 1, L + 2, \ldots \}.$$  \hfill (2.33)

As shown in Figs. 2.6(a) and 2.6(b), $L^*(m)$ is non-increasing as $m$ increases; for example, for $C_1(3, 6, m, 2)$, $L^*(1) = 15$, $L^*(3) = 10$, and $L^*(m) = 8$ when $m \geq 6$. Thus, we see that increasing the finite field size speeds up the saturation of the FSD threshold as $L$ increases. To avoid repetition, we omit the FSD thresholds obtained for other $(J,K)$-regular SC-LDPC code ensembles with varying $L$; however, it should be noted that the threshold saturation behavior described above is consistent over all considered code ensembles. In the remainder of this chapter, we adopt the convention that the coupling length is $L = 100$ unless stated otherwise.

We now consider WD thresholds of the $(3, 6, m)$ SC-LDPC code ensembles in Fig. 2.7 (for the $C_2(3, 6, m)$ and $C_1(3, 6, m, 1)$ ensembles) and 2.8 (for the $C_1(3, 6, m, 2)$
Figure 2.7. WD thresholds of the $C_2(3, 6, m)$ and $C_1(3, 6, m, 1)$ ensembles. FSD thresholds are included as benchmarks.
Figure 2.8. Continuing with Fig. 2.7: WD thresholds of the $C_1(3,6,m,2)$ and $C_1(3,6,m,3)$ ensembles.
and $C_1(3,6,m,3)$ ensembles). The WD thresholds of $C_2(3,6,m)$ with the classical edge-spreading format are shown in Fig. 2.7(a). As expected, for fixed $m$, the WD thresholds improve with increasing $W$, and we find that $\hat{\epsilon}_{WD}(m) = \epsilon^*(m)$ for $W \geq W^*(m)$, i.e., for a sufficiently large window, the WD threshold is equal to the FSD threshold for all $m$. We note that $W^*(m)$ is non-increasing as $m$ increases, i.e., the saturation of the WD thresholds $\epsilon_{WD}^*(m,W)$ to $\hat{\epsilon}_{WD}(m)$ is faster for larger $m$. For example, $W^*(2) = 15$, $W^*(4) = 12$, whereas for $m \geq 7$, $W^*(m) = 8$. Due to a combination of the existence of degree-1 variable nodes in the window and the larger coupling width $w = 2$, $C_2(3,6,m)$ does not perform well using WD with a relatively small window.

Next, we consider the cases when $w = 1$: $C_1(3,6,m,1)$, $C_1(3,6,m,2)$, and $C_1(3,6,m,3)$, shown in Figs. 2.7(b), 2.8(a), and 2.8(b), respectively. We observe that

- Similar to the $C_2(3,6,m)$ ensemble, for each of the three ensembles at a particular $m$, the WD threshold $\epsilon_{WD}^*(m,W)$ improves as $W$ increases and saturates numerically to a constant value $\hat{\epsilon}_{WD}(m)$. Again, increasing the finite field size speeds up the saturation as $W$ increases; for example, $W^*(2) = 10$, $W^*(4) = 8$, and $W^*(6) = 6$ for $C_1(3,6,m,1)$.

- Simply choosing $W \geq W^*(m)$ does not necessarily guarantee good WD thresholds, since $\hat{\epsilon}_{WD}(m)$ may not equal $\epsilon^*(m)$ even when $W$ is large. In fact, $\hat{\epsilon}_{WD}(m) = \epsilon^*(m)$ for all $m$ only for $C_1(3,6,m,1)$ and $C_1(3,6,m,2)$; for $C_1(3,6,m,3)$, on the other hand, $\hat{\epsilon}_{WD}(m)$ diverges from $\epsilon^*(m)$ as $m$ increases, as shown in Fig. 2.8(b).

We turn our attention now to the implications of the WD thresholds on protograph design. Recall the three types of edge-spreading formats of the $(3,6,m)$ SC ensembles

---

\footnote{Of course, as mentioned earlier, by selecting $W = L + w$ in WD, the decoding window covers the whole parity-check matrix and WD is equivalent to FSD. However, we do not consider this extreme case.}
with $w = 1$ defined in (2.29)-(2.31), where $B_0^1$ is given as $\begin{bmatrix} E_A & E_A \\ E_A & E_B \end{bmatrix}$, $\begin{bmatrix} E_A & E_A \\ E_A & E_B \end{bmatrix}$, and $\begin{bmatrix} E_B & E_B \end{bmatrix}$, respectively. As we move from type 1 to 2 to 3, the $q$-ary SC-LDPC code ensemble includes more $E_B = \begin{bmatrix} 2 & 1 \end{bmatrix}^T$ spreading and less $E_A = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ spreading. As illustrated in Fig. 2.9(a) for $C_1(3, 6, m, 1)$ with a window size $W = 4$, $E_A$ spreading has a strong (lower degree) check node at the beginning of the window and weak variable nodes (with degree 1) at the end of the window. As a result, for all $m$, $\hat{\epsilon}_{WD}(m) = \epsilon^*(m)$ when $W$ is large enough, but $\hat{\epsilon}_{WD}(m, W)$ is not very good when $W$ is relatively small – for example, $W = 4$ in Fig. 2.7(b). (See also the threshold behavior of the $C_2(3, 6, m)$ ensembles in Fig. 2.7(a) which have a similar structure but larger $w$.)

On the other hand, as illustrated in Fig. 2.9(b) for $C_1(3, 6, m, 3)$, $E_B$ spreading provides strong (higher degree) variable nodes at the end of the window and a weak (higher degree) check node at the beginning of the window. As a result, compared to $C_1(3, 6, m, 1)$ and $C_1(3, 6, m, 2)$, $C_1(3, 6, m, 3)$ has the smallest $W^*(m)$ when $m$ is fixed, i.e., threshold saturation to $\hat{\epsilon}_{WD}(m)$ is fastest as $W$ increases, but $\hat{\epsilon}_{WD}(m)$ itself does not converge to $\epsilon^*(m)$, resulting in unsatisfactory WD thresholds, especially when $m$ is large. In fact, comparing Fig. 2.8(b) to Fig. 2.5, we observe that the WD threshold of $C_1(3, 6, m, 3)$ becomes more “block-like” as $m$ increases, i.e., the curve for the WD threshold of $C_1(3, 6, m, 3)$ behaves similarly to the curve for the FSD threshold of $B(3, 6, m)$ for $m \geq 4$. This “block-like” behavior occurs for type 3 spreading because the edges of the block protograph have not been sufficiently spread, i.e., only one edge from each variable node in a block protograph is spread to the adjacent block protograph.

We summarize the above observations for WD thresholds with respect to the advantages and disadvantages of $E_A$ and $E_B$ spreading based on their effects on the portion of the parity-check matrix covered by the window:
Figure 2.9. The portion of the base matrix covered by the window when $W = 4$.

1. The advantage of $E_A$: Due to the strong check node at the start of the window, for a sufficiently large window size, the WD threshold saturates to the FSD threshold, which in turn approaches the channel capacity as the finite field size increases.

2. The disadvantage of $E_A$: Due to the weak variable nodes at the end of the window, WD does not perform well when the window size is relatively small, so for small finite field sizes, there are large gaps between the WD threshold and the FSD threshold.
3. The advantage of $E_B$: Due to the strong variable nodes at the start of the window, for relatively small window sizes, the WD threshold quickly saturates to its best achievable value, even for relatively small finite field sizes.

4. The disadvantage of $E_B$: Due to the weak check node at the end of the window, WD tends to provide more “block-like” behavior, so that as the finite field size increases, the WD threshold diverges from the FSD threshold of the $q$-ary SC-LDPC code ensemble and approaches the FSD threshold of the corresponding uncoupled $q$-ary LDPC-BC ensemble.

Based on the advantages and disadvantages of these two antipolar spreading formats, we can develop design rules that combine fast saturation and FSD-achieving thresholds by mixing $E_B$ spreading and $E_A$ spreading, resulting in the type 2 spreading $C_1(3, 6, m, 2)$. For example, as shown in Fig. 2.8(a), we see that $C_1(3, 6, m, 2)$ has good WD thresholds even when both $m$ and $W$ are relatively small, i.e., with $m = 5$ and $W = 5$, the best performance is already achieved and lies within 0.15% of the channel capacity. These design rules are consistent with the design rules proposed in [34] for the binary case, but they are more general in the sense that the effect of non-binary code alphabets is included.

To summarize, given the $(3, 6)$-regular degree distribution, to achieve near-capacity thresholds with small decoding latency and small decoding complexity (see Section 2.5 for details), the $q$-ary SC-LDPC code ensemble $C_1(3, 6, m, 2)$ is recommended due to its good thresholds when the WD window size and the finite field size are both relatively small.

2.3.2.3 The $(4, 8)$ and $(5, 10)$ Ensembles

We now examine the WD thresholds of the $(4, 8)$-regular $q$-ary SC-LDPC code ensembles with $w = 1$ and the $(5, 10)$-regular $q$-ary SC-LDPC code ensembles with
Figure 2.10. WD thresholds of $q$-ary SC-LDPC code ensembles with $w = 1$ and $W = 3$ from: (a) the $(3, 6, m)$ ensembles, and (b) the $(4, 8, m)$ ensembles. FSD thresholds of the corresponding $q$-ary LDPC-BC ensembles are included for reference.
Figure 2.11. Continuing with Fig 2.10: WD thresholds of $q$-ary SC-LDPC code ensembles with $w = 1$ and $W = 3$ from the $(5, 10, m)$ ensembles. Moreover, the WD thresholds of the $C_1(J, 2J, m, 2)$ ensembles, $J = 3, 4, \text{ and } 5$, with $W = 5$ are shown in (d).
$w = 1$ to explore how the advantages and disadvantages of $E_A$ spreading and $E_B$ spreading are affected by the density $(J, K)$ of the parity-check matrix, where we still have $k = K/J = 2$.

For comparison, the WD thresholds of the $(3, 6, m)$ SC ensembles with $w = 1$ and $W = 3$ are shown in Fig. 2.10(a), and the WD thresholds of the $(4, 8)$ and $(5, 10)$ SC ensembles with $w = 1$ and $W = 3$ are shown in Figs. 2.10(b) and 2.11(a), respectively. In addition to several features that are similar to the $(3, 6)$ SC ensembles, some further observations can be made for the $(4, 8)$ and $(5, 10)$ SC ensembles:

- Recall that for $E_B$ spreading, the advantage results from the strong variable nodes with degree $(J - 1)$ at the end of the window, and the disadvantage results from the weak check node with degree $2(J - 1)$ at the beginning of the window, as shown in Fig. 2.9(b) with $J = 3$. Thus, as the density $J$ increases, both the positive and the negative effects are strengthened. On the one hand, saturation of the WD threshold $\epsilon^*_\text{WD}(m, W)$ to its best achievable value $\tilde{\epsilon}_{\text{WD}}(m)$ as $W$ increases is faster. For example, for $m = 3$, we find that $W^*(3) = 4$ for $C_1(3, 6, m, 3)$, $W^*(3) = 4$ for $C_1(4, 8, m, 3)$, and $W^*(3) = 3$ for $C_1(5, 10, m, 3)$, i.e., for fixed $m$, $W^*(m)$ is non-increasing for $C_1(J, 2J, m, 3)$ as $J$ increases. On the other hand, we observe from Fig. 2.10(a), 2.10(b), and 2.11(a) that:
  
  - The WD thresholds of $C_1(J, 2J, m, 3)$ monotonically decrease as $m$ increases ($m \geq 3$ for $C_1(3, 6, m, 3)$),
  
  - Their curves are almost parallel to the corresponding curves of the FSD thresholds of $B(J, 2J, m)$ — this effect is more apparent for $J = 4$ and 5, and
  
  - The gap between the two decreases as $J$ increases.

Thus, the denser the parity-check matrix is, the more “block-like” the WD thresholds of type 3 spreading $[E_B \ E_B]$ become. As previously mentioned,
this is because only one edge from each variable node in a block protograph is spread to the adjacent block protograph in type 3 edge spreading.

- The disadvantage of $E_B$ spreading also affects the WD thresholds of type 2 spreading. Fig. 2.11(b) compares the WD thresholds of $C_1(3, 6, m, 2)$, $C_1(4, 8, m, 2)$, and $C_1(5, 10, m, 2)$ ensembles for $W = 5$. We see that, as $J$ increases, the thresholds of $C_1(J, 2J, m, 2)$ diverge more significantly from channel capacity as $m$ increases, consistent with the observation that the disadvantage of $E_B$ spreading is strengthened as $J$ increases. Moreover, the divergence occurs sooner as $J$ increases, e.g., the WD threshold of $C_1(5, 10, m, 2)$ increases only up to $m = 2$ and then starts to decrease as $m$ increases further, whereas the divergence from capacity for both $C_1(3, 6, m, 2)$ and $C_1(4, 8, m, 2)$ does not occur until $m = 6$.

- For type 1 spreading, where both columns in $B_0^1$ employ $E_A$ spreading, the WD thresholds improve dramatically as $J$ increases for small $W$, as we see in Fig. 2.10(a), 2.10(b), and 2.11(a) for $W = 3$. In other words, to a certain extent, the negative effect of $E_A$ spreading due to the presence of the degree-1 variable nodes at the end of the window, which results in poor WD thresholds for small $W$, is compensated for by the increased density of the parity-check matrix. This observation is further supported by Fig. 2.12, which compares the WD thresholds of $C_1(J, 2J, m, 1)$ and $C_1(J, 2J, m, 2)$ for $J = 3$ with $W = 5$, for $J = 4$ with $W = 5$, and for $J = 5$ with $W = 4$. We observe that for $J = 4$ and $J = 5$, $C_1(J, 2J, m, 1)$ has better thresholds than $C_1(J, 2J, m, 2)$ for all finite field sizes, even with relatively small $W$. This indicates that, although $C_1(3, 6, m, 1)$ does not perform well with WD, $C_1(4, 8, m, 1)$ and $C_1(5, 10, m, 1)$ are good $q$-ary SC-LDPC code ensembles for WD.

Based on the above observations, since the thresholds of type 2 spreading $\begin{bmatrix} E_A & E_B \end{bmatrix}$ deteriorate as $J$ increases (see Fig. 2.11(b)), while the thresholds of type 1 spreading...
Figure 2.12. Comparison of WD thresholds: type 1 spreading vs. type 2 spreading for $J = 3$ with $W = 5$, for $J = 4$ with $W = 5$, and for $J = 5$ with $W = 4$. Note that for the latter two comparisons, $C_1(J, 2J, m, 1)$ has better thresholds than $C_1(J, 2J, m, 2)$ for all $m$.

\[
\begin{bmatrix}
E_A & E_A \\
E_A & E_B
\end{bmatrix}
\]

improve, we conclude for these two edge-spreading formats that

1. When $J = 3$, $C_1(3, 6, m, 2)$ is good for WD, compared to $C_1(3, 6, m, 1)$,

2. When $J = 4$, both $C_1(4, 8, m, 1)$ (for all $m$) and $C_1(4, 8, m, 2)$ (for $m \leq 6$) are good for WD, and

3. When $J = 5$, $C_1(5, 10, m, 1)$ is a better choice for WD than $C_1(5, 10, m, 2)$.

Moreover, for $J = 3$, if the code construction is restricted to use a very small finite field size – say $m = 1$ ($q = 2^m = 2$) or $m = 2$ ($q = 4$) – and the requirement for threshold can be slightly relaxed, then $C_1(3, 6, m, 3)$ with the $\begin{bmatrix} E_B & E_B \end{bmatrix}$ spreading is good for WD as well (see Fig. 2.8(b)). Again, this is consistent with the design rules proposed in [34] for the binary SC-LDPC code ensembles suitable for WD. Nevertheless, $C_1(4, 8, m, 3)$ and $C_1(5, 10, m, 3)$ ensembles are obviously not good for WD, as shown in Fig. 2.10(b) and 2.11(a).
As the key idea throughout this chapter, we emphasize that desirable protograph-based $q$-ary SC-LDPC code ensembles for windowed decoding should achieve good thresholds when both the finite field size and the window size are small. To this end, we can summarize the above observations made for the $(3, 6)$, $(4, 8)$, and $(5, 10)$ SC ensembles with $w = 1$ into design rules as follows:

- Combining $E_A$ spreading and $E_B$ spreading, i.e., type 2 spreading, is attractive when $J$, characterizing the density of the parity-check matrix, is small;

- As $J$ increases, $E_B$ spreading becomes less attractive, and it should be avoided in favor of $E_A$ spreading when $J \geq 5$.

The classical $(4, 8)$-regular and $(5, 10)$-regular $q$-ary SC-LDPC code ensembles with $w = J - 1$, i.e., $C_3(4, 8, m)$ and $C_4(5, 10, m)$ defined by (2.28), provide WD thresholds analogous to $C_2(3, 6, m)$. To be more specific, for all $m$, the WD thresholds $\epsilon^*_WD(m, W)$ improve with $W$ and saturate to $\hat{\epsilon}_{WD}(m) = \epsilon^*(m)$ (the FSD thresholds), which are non-decreasing and numerically achieve capacity as $m$ increases. Further, when $m$ is fixed and $W$ is sufficiently large, the WD thresholds of the $C_J^{-1}(J, 2J, m)$ ensembles improve as $J$ increases, as shown in Fig. 2.13 for $m = 2$, $W = 8$ and 10. Nevertheless, when $W$ is small to moderate, the thresholds are not satisfactory; for example, when $W = 6$, there is still a significant space for threshold improvement by increasing $W$ further. In fact, since the minimum $W$ required for WD of $C_{J-1}(J, 2J, m)$ is $(w+1) = J$, as $J$ increases the classical edge spreading format (2.28) is even less attractive if there is a constraint on decoding latency, i.e., if a small $W$ must be adopted. For example, $\epsilon^*_WD(m, W = 4) = 0$ for $C_4(5, 10, m)$, as shown in Fig. 2.13, because the minimum required window size in this case is $W = 5$. 
Figure 2.13. WD thresholds of $\mathcal{C}_{J-1}(J, 2J, m = 2)$ ensembles as $J$ increases for window sizes $W = 4, 6, 8,$ and $10$. (For $\mathcal{C}_4(5, 10, m)$ with $W = 4$, $\epsilon_{WD}^*(m, 4) = 0$, because the minimum required $W$ is 5.)

2.3.3 Numerical Results: $k = 3$ and $k = 4$ ($R > 1/2$)

The previous subsection presented the advantages and disadvantages of using $E_A$ spreading and $E_B$ spreading in the construction of rate-$R = 1/2$ protograph-based $q$-ary SC-LDPC code ensembles suitable for WD, and results were presented on the influence of varying the density $J$ (and thus $K$) of the parity-check matrix on the WD thresholds. This subsection presents additional results on WD thresholds for higher rate ($R = 2/3$ and $3/4$) protograph-based $q$-ary SC-LDPC code ensembles, with emphasis on how the particular mix of $E_A$ spreading and $E_B$ spreading affects the WD thresholds. We expect that the more a certain kind of spreading is used, the more its corresponding advantages and disadvantages will be observed. For simplicity, we fix $J = 3$. 

47
2.3.3.1 \((J, K) = (3, 9), k = 3\)

We consider the \((3, 9)\) ensembles over \(GF(2^m)\) defined by (2.13). The asymptotic rate of \((3, 9)\)-regular \(q\)-ary SC-LDPC code ensembles is \(R = (k - 1)/k = 2/3\), when the coupling length \(L\) goes to infinity. Since \(k = 3\), the component matrices \(B_i\) used to construct \(B_{SC}\) in (2.2) are of size \(1 \times 3\) and, in addition to the classical edge spreading with \(w = 2\), there are four types of \(w = 1\) spreading where, for types 1 through 4, \(B_0^1\) is given as

\[
\begin{bmatrix}
E & A & E \\
E & A & E \\
E & B & E \\
E & B & E
\end{bmatrix}, \quad \begin{bmatrix}
E & A & E \\
E & A & E \\
E & B & E \\
B & E & E
\end{bmatrix}, \quad \begin{bmatrix}
E & A & E \\
A & E & E \\
B & E & E \\
E & B & E
\end{bmatrix}, \quad \begin{bmatrix}
E & B & E \\
E & B & E \\
E & B & E \\
B & E & E
\end{bmatrix},
\]

respectively.

We expect that if there are more \(E_B\) spreadings, then for fixed \(m\), the WD threshold \(\epsilon^*_WD(m, W)\) will saturate faster to the best achievable value \(\hat{\epsilon}_{WD}(m)\) as \(W\) increases, and that if there are more \(E_A\) spreadings, then \(\hat{\epsilon}_{WD}(m)\) will diverge less from channel capacity as \(m\) increases. These expectations are met, as illustrated in Figs. 2.14(a) and 2.14(b) when \(W = 4\) and 10, respectively. Combined with other numerical results, it is observed that

- When \(m\) is fixed, \(\epsilon^*_WD(m, W)\) of \(C_1(3, 9, m, 4)\), which contains all \(E_B\) spreadings, has the fastest saturation to the corresponding \(\hat{\epsilon}_{WD}(m)\) among the WD thresholds of all the \(w = 1\) code ensembles. This indicates that there is little space for threshold improvement for \(C_1(3, 9, m, 4)\) by increasing \(W\) to a large value; indeed, if comparing the two \(C_1(3, 9, m, 4)\) curves in Fig. 2.14(a) and 2.14(b), we observe that, over the entire range of \(m\), \(\epsilon^*_WD(m, W)\) hardly changes when the window size goes from as small as \(W = 4\) to a sufficiently large \(W = 10\). In fact, even in the extreme case \(m = 1\) with the slowest saturation \((W^*(m) = 6)\) among all field sizes, \(\epsilon^*_WD(m, 4)\) for \(C_1(3, 9, m, 4)\) already lies within 0.35% of \(\hat{\epsilon}_{WD}(m)\).

- On the other hand, this fast saturation of \(\epsilon^*_WD(m, W)\) to \(\hat{\epsilon}_{WD}(m)\) is accompanied by reduced threshold values. In Fig. 2.14(a), where \(m \leq 3\) and \(W\) is small,
Figure 2.14. WD thresholds of the (3, 9, m) SC ensembles with \( w = 1 \): (a) \( W = 4 \), and (b) \( W = 10 \), a sufficiently large window size such that the best WD thresholds are achieved for all the SC-LDPC code ensembles. The FSD thresholds of \( B(3, 9, m) \) are included as a benchmark.
the ordering of the ensemble types from best to worst is 4, 3, 2, 1 – WD of $C_1(3, 9, m, 1)$ has even worse performance than FSD of the block ensemble $B(3, 9, m)$. When $W < W^*(m)$, the WD thresholds are worse than $\hat{\epsilon}_{WD}(m)$ because the decoder performance is impaired. In this regime, any additional structural weakness, such as weak variable nodes arising from an $E_A$ spreading, further harm the threshold, especially at a small $m$, where we observe that fewer $E_A$ spreadings result in better thresholds. However, for a larger window size, the decoder is more robust, in the sense that some weaker variable nodes can be included without significantly harming the decoder performance. This allows for a stronger check node at the start of the window that initiates the “wave-like” decoding and threshold saturation of SC-LDPC code ensembles. This effect is more obvious in Fig. 2.14(b), where the window size $W = 10$ is chosen to be sufficiently large such that $\epsilon^*_WD(m, 10) = \hat{\epsilon}_{WD}(m)$ for all the code ensembles, i.e., $W = 10 \geq W^*(m)$. Compared to Fig 2.14(a), for $m \leq 3$, types 1, 2, and 3 now provide almost identical WD thresholds, which are better than type 4. In this regime, we clearly wish to pick an edge spreading format with a mixture of $E_A$ and $E_B$.

- The introduction of $E_B$ spreadings causes a divergence from capacity $\epsilon_{Sh} = 1/3$ of a rate-$R = 2/3$ code ensemble as $m$ increases. This is observed no matter the window size is small (Fig. 2.14(a)) or large (Fig. 2.14(b)), and the divergence becomes more significant as more $E_B$ spreadings are used. This behavior is similar to what was observed for $C_1(3, 6, m, 3)$ in Fig. 2.10(a), and again, the “block-like” behavior as $m$ increases can be explained by an insufficient edge spreading.

- Finally, the $C_1(3, 9, m, 1)$ ensembles, with all $E_A$ spreading, and the $C_2(3, 9, m)$ ensembles display non-decreasing maximum WD thresholds $\hat{\epsilon}_{WD}(m)$ that ap-
proach channel capacity as $m$ increases. However, the weak variable nodes at the end of the windows for these two ensembles imply that when $m$ is small, $W^*(m)$ should be large.

2.3.3.2 $(J, K) = (3, 12), ~ k = 4$

The asymptotic rate of $(3, 12)$-regular $q$-ary SC-LDPC code ensembles is $R = (k - 1)/k = 3/4$, when the coupling length $L$ goes to infinity. For $w = 1$ and an arbitrary $m$, there are $k+1 = 5$ types of $q$-ary SC-LDPC code ensembles in (2.13), and the behavior of their thresholds is similar to the $(3, 9, m)$ SC ensembles with $w = 1$. Fig. 2.15(a) shows the percentage divergence of the $\hat{\epsilon}_{\text{WD}}(m)$ from the corresponding channel capacities for the $C_1(3, 6, m, 3)$, $C_1(3, 9, m, 4)$, and $C_1(3, 12, m, 5)$ ensembles, where all $E_B$ spreading is adopted in each case. This comparison strengthens our observation that the more a particular spreading is used, the greater are its effects: the WD threshold of $C_1(3, 12, m, 5)$ shows the most significant divergence from the corresponding BEC capacity because it uses four $E_B$ spreadings, compared to three in $C_1(3, 9, m, 4)$ and two in $C_1(3, 6, m, 3)$.

Similar observations can be made in Fig. 2.15(b) as well, which compares the percentage divergence for the $C_1(3, 6, m, 2)$, $C_1(3, 9, m, 3)$, and $C_1(3, 12, m, 4)$ ensembles, where one $E_A$ spreading and $(k - 1)$ $E_B$ spreadings are adopted in each case. Again, the ensemble uses the most $E_B$ spreadings – in this case, $C_1(3, 12, m, 4)$ with three $E_B$’s – shows the most significant divergence as $m$ increases. However, compared to the thresholds in Fig. 2.15(a) with the same degree distribution $(J = 3, K = 3k)$, we observe that even introducing only one $E_A$ spreading can significantly alleviate the divergence effect from the $E_B$ spreading(s) and thus improve the WD thresholds, i.e., mixing $E_A$ and $E_B$ is promising for design of WD-suitable code ensembles.

The classical edge-spreading for the $C_{J-1}(J, 3J, m)$ code ensembles is not suitable for WD, despite their excellent capacity-achieving thresholds when $m$ and $W$ are
Figure 2.15. Percentage divergence of the best achievable WD thresholds $\hat{\epsilon}_{WD}(m)$ from the corresponding channel capacities for (a) the $C_1(3, 6, m, 3)$, $C_1(3, 9, m, 4)$, and $C_1(3, 12, m, 5)$ ensembles with all $E_B$ spreading, and (b) the $C_1(3, 6, m, 2)$, $C_1(3, 9, m, 3)$, and $C_1(3, 12, m, 4)$ ensembles containing only one $E_A$ spreading.
both large enough; this is similar to the $C_{J-1}(J, 2J, m)$ code ensembles.

Finally, we emphasize again the design rule that combining $E_B$ spreading and $E_A$ spreading is a good strategy for designing $(J, K)$-regular $q$-ary spatially coupled LDPC code ensembles suitable for windowed decoding when $J$ is small for two reasons:

1. The coupling width $w = 1$ makes the minimum required window size only $W = w + 1 = 2$, and

2. The threshold can be near capacity when $m$ and $W$ are both small.

The above conclusions are supported by WD thresholds for the $C_1(3, 6, m, 2)$, $C_1(4, 8, m, 2)$, $C_1(3, 9, m, 2)$, $C_1(3, 9, m, 3)$, $C_1(3, 12, m, 2)$, and $C_1(3, 12, m, 3)$ ensembles. For the cases when $J = 3$, these conclusions are further reinforced by decoding performance simulations of finite-length codes with different rates; see [37] for details.

2.4 Threshold Analysis of $q$-ary SC-LDPC Code Ensembles on the BIAWGNC

2.4.1 $q$-ary Protograph EXIT Analysis on the BIAWGNC

We use the $q$-ary protograph EXIT (PEXIT) algorithm presented in [10] to analyze the FSD thresholds of protograph-based $q$-ary SC-LDPC code ensembles on the BIAWGNC, assuming that the binary image of a codeword is transmitted using BPSK modulation, and extend it in a similar fashion to the $q$-ary WD-PDE algorithm to obtain WD thresholds. Similar to the $q$-ary PDE analysis on the BEC, the $q$-ary PEXIT analysis is also a BP algorithm performed on a protograph, where the messages now represent mutual information (MI) values, a model obtained by approximating the distribution of the log-likelihood ratio messages in BP decoding as (jointly) Gaussian. The thresholds are obtained by determining the smallest signal-to-noise ratio $E_b/N_0$ (in dB) for which decoding is successful, i.e., the smallest value of
$E_b/N_0$ such that the \textit{a-posteriori} MI between each variable node and a corresponding codeword symbol goes to 1 as the number of iterations goes to infinity.

2.4.2 Numerical Results

Our observations and conclusions made regarding the WD thresholds of $q$-ary SC-LDPC code ensembles on the BIAWGNC are similar to those made for the BEC. As a result, only a few examples are given here.

Fig. 2.16(a) compares the FSD thresholds of the $(2, 4, m)$ and $(3, 6, m)$ ensembles on the BIAWGNC.\footnote{Due to computational limitations, the BIAWGNC thresholds were calculated only up to $m = 8$. However, as stated by Uchikawa \textit{et al.} in [31], it is reasonable to assume that the BIAWGNC thresholds for $m = 9$ and 10 are consistent with the corresponding BEC results.} $C_1(3, 6, m, 3)$ and $C_1(3, 6, m, 1)$ have the same FSD thresholds for all $m$, which are almost identical to those of $C_1(3, 6, m, 2)$, so only $C_1(3, 6, m, 2)$ is shown to represent the $w = 1$ code ensembles and to compare with $C_2(3, 6, m)$. Fig. 2.16(b) shows the WD thresholds of $C_1(3, 6, m, 2)$ when $W = 3$ and 5. Both subfigures illustrate behavior similar to the BEC results presented in Section 2.3.2. To summarize, small gains are observed for $C_1(2, 4, m)$ compared to $B(2, 4, m)$ until the finite field size gets large, whereas (numerically) capacity achieving WD thresholds that are significantly better than the corresponding block code thresholds are observed for both $C_1(3, 6, m, 2)$ and $C_2(3, 6, m)$. Again, $C_1(3, 6, m, 2)$ turns out to be particularly well suited for WD; for $m = 5$ and $W = 5$, the WD threshold is essentially at capacity.

2.5 Decoding Latency and Decoding Complexity

This section considers the tradeoff between two critical decoding properties for the implementation of $q$-ary spatially coupled LDPC code ensembles:

1. \textit{Latency}: measured as the number of bits that must be received before decoding
Figure 2.16. FSD thresholds of the $(2, 4, m)$ and $(3, 6, m)$ ensembles and WD thresholds of $C_1(3, 6, m, 2)$ on the BIAWGNC.
can begin, and

2. *Complexity*: measured as the number of decoding operations required per information bit.

Our focus is the ensemble average behavior on the BEC when windowed decoding is used; different cases are compared on the same BEC, so that the tradeoff between decoding latency and decoding complexity can be examined. We use the $q$-ary WD-PDE algorithm (for WD) and the $q$-ary PDE algorithm (for FSD for comparison) in order to obtain our results, i.e., we consider an infinite lifting factor $M$ used for the ensemble construction, thereby removing the effect of $M$ from the latency-complexity tradeoffs. This allows us to get a general picture of the latency-complexity tradeoffs associated with a particular code ensemble, rather than analyzing specific, finite-length codes. This section is intended to guide the design of practical, finite-length protograph-based $q$-ary SC-LDPC codes, especially when there is a limit on decoding latency.

In the remainder of this section, we focus on the $(J,2J)$ SC ensembles with the coupling width $w = 1$, previously discussed in Section 2.3.2, to simplify the discussion; however, similar results can be obtained for other code ensembles as well.

2.5.1 Decoding Latency

For a $q$-ary SC-LDPC code constructed as described in Section 2.2.3, the decoding latency (normalized by $M$) of WD is given by $kmW$ measured in *bits*; recall that the binary image of a codeword is transmitted, so each GF($q$) symbol contains $m$ bits. Since $k = 2$, the latency is proportional to the product of $m$ and $W$. Thus, in the numerical results presented in this section, we use

$$T_{SC} = mW \quad (2.34)$$
to represent the latency of WD for a $q$-ary SC-LDPC code ensemble. And FSD of the ensemble can be viewed as a special case of WD for which $W = L + w$, where $L$ is the coupling length, so the corresponding latency is also calculated by (2.34).

### 2.5.2 Decoding Complexity

As stated in [7] and the references therein, the decoding complexity of $q$-ary LDPC codes using the sum-product algorithm based on the fast Fourier transform can be summarized as follows:

- One check-to-variable update requires $O(q \log q)$ operations, and
- One variable-to-check update requires $O(q)$ operations.

Thus we use “$q \log_2 q$” and “$q$” to represent the complexity of c-to-v and v-to-c updates, respectively, for decoding on the BEC.

We define decoding complexity as the number of operations required per information bit, which is a fraction $1/(R_L m k M L)$ of the total number of operations for all the c-to-v and v-to-c updates during the decoding process, and $R_L$ is the design rate. That is,

$$\text{Decoding complexity} = \frac{J (q + q \log_2 q) k M \sum_{t=1}^{L} l_t}{R_L m k M L} = \frac{J 2^m (m + 1) \sum_{t=1}^{L} l_t}{R_L m L},$$

where $l_t$ is the number of iterations involving updates of variables at time instant $t$ ($1 \leq t \leq L$), which can be easily tracked during the decoding process. As previously mentioned, we let $L = 100$.

Although (2.35) is derived for BP decoding of finite-length SC-LDPC codes, we use it for our ensemble-average complexity analysis as well. For a particular $q$-ary SC-LDPC code ensemble, the erasure rate of the BEC is chosen to be no greater than the WD threshold of the ensemble, so $q$-ary (WD-)PDE is guaranteed to declare a
decoding success. As the algorithm iterates, the number of updates of check nodes and variable nodes at each time instant is tracked via $t$, and then the decoding complexity is calculated.

2.5.3 Numerical Results

2.5.3.1 WD vs. FSD, with the Same Decoding Threshold

Fig. 2.17 uses $\mathcal{C}_1(3, 6, m, 2)$ as an example to illustrate why WD is preferred to FSD of $q$-ary SC-LDPC code ensembles.

For each $m$, FSD is compared to WD with $W = W^*(m)$, where we recall that $W^*(m)$ is the minimum window size that provides the best achievable WD threshold $\hat{\epsilon}_{WD}(m)$; here, $W^*(m) = 10, 8, 6, 5, 4, 4, 4, 4, 4, 4$ for $m = 1$ to 10. Since $\hat{\epsilon}_{WD}(m) = \epsilon^*(m)$ for $\mathcal{C}_1(3, 6, m, 2)$ for all $m$, these two cases have the same decoding threshold, and the channel erasure rate is set to this threshold, i.e., $\epsilon = \epsilon^*(m)$. From Fig. 2.17(a), we see that WD saves approximately 75% to 90% in operations compared to FSD, as $m$ ranges from 1 to 10; the larger the finite field size, the more operations are saved.

Moreover, as shown in Fig. 2.17(b), WD also has a significant advantage in reducing decoding latency: the decoding latency of WD is only about 10% of that of FSD when $m = 1$ ($q = 2$) and only about 4% when $m = 10$ ($q = 1024$) – i.e., the larger the finite field size, the more decoding latency is saved.

To summarize, WD is preferred to FSD for decoding $q$-ary SC-LDPC codes, because the former provides large savings in both decoding complexity and decoding latency due to the fact that, unlike FSD, WD is localized to include only a small portion of the parity-check matrix. Also, by choosing the window size appropriately, these savings incur only a negligible loss in threshold.
Figure 2.17. WD vs. FSD for $C_1(3, 6, m, 2)$: comparison of (a) decoding complexity, and (b) decoding latency.
2.5.3.2 WD Complexity as a Function of \(m\) and \(W\), with Equal Latency

Decoding latency is represented by \(mW\), so if \(mW\) is fixed, there can be multiple \((m,W)\) pairs that satisfy an equal-latency constraint. Again, using the \(C_1(3,6,m,2)\) ensemble as an example, Tables 2.1 and 2.2 show the decoding complexity of different \((m,W)\) pairs, when \(mW\) is fixed at 12, 20, 24, 30, 40, and 60; the third column is for a BEC with \(\epsilon = 0.488\), while the fourth column is for \(\epsilon = 0.44\). For each \(\epsilon\), the smallest decoding complexity for a particular \(mW\) value is marked in boldface, corresponding to the most attractive \((m,W)\) pair for that particular decoding latency.

The channel erasure rate \(\epsilon = 0.488\) is within approximately 0.1\% of the best-achievable binary WD threshold of \(C_1(3,6,m,2)\) and 2.5\% from channel capacity.\(^7\) As a result, when \(m = 1\), a large number of iterations is required to achieve decoding success using the \(q\)-ary DE algorithm. On the other hand, larger values of \(m\) (and as a result, smaller values of \(W\)), for example, \(m = 2, 3,\) and 4, show significant reductions in decoding complexity (one to two orders of magnitude), since the WD thresholds for the corresponding \((m,W)\) pairs are larger; in fact, the smallest decoding complexity is achieved when \(m\) is either 2 or 3 for all the decoding latencies examined in Tables 2.1 and 2.2.

\(q\)-ary SC-LDPC codes may still provide benefits compared to their binary counterparts even at lower channel erasure rates. For example, in the fourth column of Tables 2.1 and 2.2, \(\epsilon = 0.44\) is approximately 10\% from the best achievable binary WD threshold of \(C_1(3,6,m,2)\) and 12\% from channel capacity. Here, we see that \(m = 2\) outperforms \(m = 1\) for all decoding latencies and achieves the smallest decoding complexity in all cases, although the performance gain compared to \(\epsilon = 0.488\) is not large. Eventually, the advantage of 4-ary codes compared to binary codes disap-

\(^7\)For a fixed value of \(mW\), not all possible \((m,W)\) pairs can guarantee decoding success for this channel erasure rate. For example, when \(mW = 12, m = 3\) and \(W = 4\) results in a threshold below \(\epsilon = 0.488\).
pears as $\epsilon$ decreases further; nevertheless, Tables 2.1 and 2.2 suggest that, in order to satisfy stringent performance requirements with a constraint on decoding latency, one should consider $q$-ary SC-LDPC codes as alternatives to binary codes.
### TABLE 2.1

**DECODING COMPLEXITY OF $C_1(3, 6, m, 2)$**

<table>
<thead>
<tr>
<th>Decoding Latency $mW$</th>
<th>$(m, W)$</th>
<th>Decoding complexity $\epsilon = 0.488$</th>
<th>Decoding complexity $\epsilon = 0.44$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>(1, 12)</td>
<td>$4.55 \times 10^5$</td>
<td>$1.14 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(2, 6)</td>
<td>$9.25 \times 10^3$</td>
<td>$1.03 \times 10^3$</td>
</tr>
<tr>
<td>20</td>
<td>(1, 20)</td>
<td>$7.18 \times 10^5$</td>
<td>$1.77 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(2, 10)</td>
<td>$1.32 \times 10^4$</td>
<td>$1.37 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(4, 5)</td>
<td>$1.54 \times 10^4$</td>
<td>$2.77 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(5, 4)</td>
<td>$2.62 \times 10^4$</td>
<td>$4.92 \times 10^3$</td>
</tr>
<tr>
<td>24</td>
<td>(1, 24)</td>
<td>$8.38 \times 10^5$</td>
<td>$2.07 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(2, 12)</td>
<td>$1.57 \times 10^4$</td>
<td>$1.62 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(3, 8)</td>
<td>$1.36 \times 10^4$</td>
<td>$2.05 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(4, 6)</td>
<td>$1.76 \times 10^4$</td>
<td>$3.07 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(6, 4)</td>
<td>$4.43 \times 10^4$</td>
<td>$8.96 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>(8, 3)</td>
<td>$2.06 \times 10^5$</td>
<td>$3.12 \times 10^4$</td>
</tr>
</tbody>
</table>
### Table 2.2

Continuing with Table 2.1: Decoding Complexity of $\mathcal{C}_1(3, 6, m, 2)$

<table>
<thead>
<tr>
<th>Decoding Latency $mW$</th>
<th>$(m, W)$</th>
<th>Decoding complexity</th>
<th>$\epsilon = 0.488$</th>
<th>$\epsilon = 0.44$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>(1, 30)</td>
<td>$1.00 \times 10^6$</td>
<td>2.49 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2, 15)</td>
<td>$1.92 \times 10^4$</td>
<td>$1.98 \times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3, 10)</td>
<td>$1.68 \times 10^4$</td>
<td>2.51 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5, 6)</td>
<td>$3.23 \times 10^4$</td>
<td>5.91 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6, 5)</td>
<td>$5.15 \times 10^4$</td>
<td>9.77 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10, 3)</td>
<td>$5.15 \times 10^5$</td>
<td>1.12 $\times 10^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1, 40)</td>
<td>$1.23 \times 10^6$</td>
<td>3.09 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2, 20)</td>
<td>$2.48 \times 10^4$</td>
<td>$2.55 \times 10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4, 10)</td>
<td>$2.84 \times 10^4$</td>
<td>4.88 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5, 8)</td>
<td>$4.24 \times 10^4$</td>
<td>7.75 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8, 5)</td>
<td>$1.91 \times 10^5$</td>
<td>3.67 $\times 10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10, 4)</td>
<td>$5.85 \times 10^5$</td>
<td>1.13 $\times 10^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1, 60)</td>
<td>$1.54 \times 10^6$</td>
<td>3.94 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2, 30)</td>
<td>$3.46 \times 10^4$</td>
<td>$3.58 \times 10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3, 20)</td>
<td>$3.16 \times 10^4$</td>
<td>4.71 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4, 15)</td>
<td>$4.13 \times 10^4$</td>
<td>7.10 $\times 10^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5, 12)</td>
<td>$6.21 \times 10^4$</td>
<td>1.13 $\times 10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6, 10)</td>
<td>$9.95 \times 10^4$</td>
<td>1.88 $\times 10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10, 6)</td>
<td>$8.63 \times 10^5$</td>
<td>1.66 $\times 10^5$</td>
<td></td>
</tr>
</tbody>
</table>
2.5.3.3 $\mathcal{B}(J, 2J, m)$ vs. $\mathcal{C}_1(J, 2J, m, 3)$, with Equal Latency

We now compare the WD complexity of the $\mathcal{C}_1(J, 2J, m, 3)$ ensembles (with $B_0 = \begin{bmatrix} J - 1 & J - 1 \end{bmatrix}$ and $B_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}$) with $W = 2 = w + 1$ to the FSD complexity of the $\mathcal{B}(J, 2J, m)$ ensembles defined by

$$B = \begin{bmatrix} B_0 & B_1 \\ B_1 & B_0 \end{bmatrix}.$$  \hspace{1cm} (2.36)

Similar to the derivation of (2.34), the FSD (normalized) latency of $\mathcal{B}(J, 2J, m)$ is

$$T_{BC} = 2m,$$  \hspace{1cm} (2.37)

the same as $T_{SC} = mW = 2m$ here for $\mathcal{C}_1(J, 2J, m, 3)$, i.e., the decoding latency is equal.

The decoding complexities for $J = 3$ are illustrated in Fig. 2.18. For each $m$, the channel erasure rate is chosen as the FSD threshold $\epsilon^*$ of $\mathcal{B}(3, 6, m)$, which is smaller than the WD threshold $\epsilon_{WD}(m, W)$ of $\mathcal{C}_1(3, 6, m, 3)$ with $W = 2$. Similar results can be obtained for $J = 4$ and $J = 5$ as well. For $w = 1$, using $W = 2$ results in the smallest possible decoding latency, so Fig. 2.18 suggests that, even under a very tight latency constraint, $q$-ary SC-LDPC code ensembles with type 3 spreading still provide a significant reduction in decoding complexity compared to their block counterparts.

For a comparison of finite-length $q$-ary SC-LDPC codes and $q$-ary LDPC-BCs, where the lifting factor $M$ can be varied to achieve various tradeoffs in error probability, complexity, and latency, we refer the reader to [37].

2.6 Summary

This chapter proposes design rules for $q$-ary spatially coupled LDPC codes suitable for latency-constrained applications. The design rules are based on an analysis of the
windowed decoding thresholds of various protograph-based $(J,K)$-regular $q$-ary SC-LDPC code ensembles for both the binary erasure channel and the BPSK-modulated additive white Gaussian noise channel. In particular, we show that mixing $E_A$ and $E_B$ edge spreadings to construct $q$-ary SC-LDPC code ensembles results in near-capacity WD thresholds when both the finite field size $m$ and the window size $W$ are relatively small, and that the balance between these two types of spreading depends on the degree distribution and the threshold requirements.

By tracking the number of density evolution update operations needed for decoding success of a $q$-ary SC-LDPC code ensemble for fixed channel conditions, we also demonstrate that WD is superior to FSD in both decoding complexity and decoding latency. Finally, when operation close to the binary SC-LDPC code ensemble threshold is required, we show that codes from $q$-ary SC-LDPC code ensembles provide significant reductions in decoding complexity compared to binary codes for the same
decoding latency.
CHAPTER 3

BANDWIDTH-EFFICIENT COOPERATIVE COMMUNICATION

3.1 Introduction

Wireless communication links are vulnerable to multipath fading. A common way to ameliorate the effects of such fading is through the introduction of diversity, i.e., the exploitation of multiple channel realizations to increase the likelihood that some realized channel is good enough to carry the required information. Different diversity techniques – time, frequency, spatial, angle-of-arrival, polarization, etc. [38] – have demonstrated efficacy in mitigating the effects of fading; among them, spatial diversity, present in multi-antenna systems, is the focus of this chapter.

The last decade has witnessed the emergence of cooperative communications [39], a paradigm in which single-antenna nodes share their resources and coordinate with one another to generate a virtual antenna array. This enables the robustness of a multi-antenna system while satisfying the size/power constraints typically required of mobile units. As a special case of spatial diversity, cooperative diversity has been shown to be effective [40], and the integration of cooperative communications into wireless systems has attracted the interest of several IEEE task groups (see [41] and the references therein) and other organizations [42].

A common example is illustrated in Figure 3.1. Two source/partner nodes, A and B, “help” each other to convey their data to a common destination D. We assume that the transmission time is slotted, and Node A transmits during the first half of each time slot, while Node B transmits during the second half. Due to the broadcast
Figure 3.1. Node A and B cooperate to convey their data to a common destination Node D.

nature of the wireless medium, a silent Node B, for instance, can overhear Node A’s transmission and relay appropriately the next time when it is Node B’s turn to transmit. Since different links are (typically) independent, Node A’s data thus has two independent paths to Node D, one direct and the other through Node B. In this way, diversity is achieved when Node D attempts to recover Node A’s data.

A critical aspect of this scenario is the “dual” role of each transmitter. For example, when Node A transmits, it must dedicate some of its resources to transmitting its own locally-generated data and some to relaying Node B’s data – assuming that Node A was able to recover that data. Towards this end, the classic approach is to create two orthogonal channels, one to transmit local data, the other to transmit relayed data. In [39], these orthogonal channel are created via time sharing, i.e., each partner first transmits only its local data and then transmits only the relayed data, while in [40] the orthogonal channels are created using direct sequence spread spectrum signaling. In contrast, in [43], non-orthogonal channels are formed by the Euclidean space superposition of two binary pulse amplitude modulation signals.

A related technique is the combination of channel coding and relaying called
coded cooperation, as proposed in [44]. In coded cooperation, the source pair works together in a repeating two-phase protocol: In the first phase, each source encodes its $K$ local data bits onto an $N_1$-bit codeword and transmits those $N_1$ bits. If a source was able to successfully decode its partner’s $K$ data bits from that first-phase transmission, it then computes an additional $N_2$ parity bits to augment its partner’s transmission, and it transmits those $N_2$ bits during the second phase. On the other hand, if the source was unable to decode its partner’s $K$ data bits, then during phase two it simply generates an additional $N_2$ parity bits to augment its own phase-one transmission. After both phases are complete, each partner has transmitted $(N_1 + N_2)$ bits. Assuming a reliable partner-to-partner link, each partner transmits $N_1$ bits derived from its own local data and $N_2$ bits derived from its partner’s data. The capacity advantage of this over repetition coding (in which the partner simply repeats what it has decoded rather than computing additional parities) is observed in [44].

One shortcoming of the approaches in [39], [40], [43] and [44] is the (relatively) inefficient way resources are applied to the partner-to-partner link. In each case, when a source node is acting as a relay, it transmits with only the destination D as its intended receiver; none of the resources expended during the relaying operation helps the partner-to-partner link. To improve efficiency, other techniques (for example, references [45] and [46]) have been developed in which local and relayed information are combined in ways that enable each partner’s knowledge of the data being relayed by the other partner to be exploited in a way that improves the partner-to-partner link.

In [45], local and relayed data are encoded and then XORed together prior to modulation, thus carrying out algebraic superposition of channel codes over the binary field. At the receiving partner, the relayed sequence is treated as a known “scrambling pattern” and canceled prior to decoding, while at the destination the
decoder operates to recover both local and relayed data. In [46], local and relayed information are first multiplexed together and then convolutionally encoded into one encoded sequence. At the receiving partner, trellis-pruning is employed, i.e., state transition edges that are inconsistent with the known, relayed data are deleted from the trellis, increasing the probability of decoding success. Note that both [45] and [46] exploit the fact that, for instance, Node B knows what Node A is relaying because the relayed data originated at Node B – knowledge that none of the previous approaches exploited. These approaches offer approximately 3.0 to 4.0 dB of coding gain compared to the time-sharing approach.

All of the above-referenced work assumes binary signaling\(^1\). Regarding uncoded bandwidth-efficient (i.e., QAM/PSK) transmission, reference [47] uses the in-phase component of each QAM symbol to convey local data and the quadrature component to convey relayed data. This approach, like the others above, still does not exploit each partner's knowledge of what the other is relaying, and from the perspective of the partner-to-partner link, the quadrature component of the transmitted signal is “wasted.” Reference [48] overcomes this problem to some extent by using set-partition labeling [2] to map uncoded local and relayed data onto QAM symbols. Finally, reference [49] discusses design issues for cooperative multiple trellis coded modulation schemes, in which the signal constellation is asymmetric and multiple mappings are adopted among users. Examples of using turbo coding, low-density parity-check (LDPC) coding, lattice coding, etc. in bandwidth-efficient cooperation are also briefly summarized in [49].

This chapter describes new approaches to cooperative coded modulation based on bit-interleaved coded modulation and mapping via set partitioning. The use of powerful binary codes together with a QAM/PSK mapper that enables each partner

\(^1\)Reference [43] combines two binary signals at different power levels, so the result is a skewed 4-ary modulation scheme.
to exploit its knowledge of what the other partner is relaying provides a substantial
coding gain over previous approaches. Both simulations and outage results are used
to illustrate the efficacy of the new approach.

3.2 System Description and General Approaches

Recall that the transmission time is slotted, and during time slot \( t \), Nodes A and
B each generate \( k \) bits of local information labeled \( i_A^L(t) \) and \( i_B^L(t) \). Node A transmits
in the first half slot, followed by Node B in the second half. When it is Node A’s turn
to transmit, it conveys \( i_A^L(t) \); moreover, if Node A was able to successfully decode
\( i_B^L(t-1) \), then it conveys that as well. Node B acts analogously during time slot
\( t \) – conveying \( i_B^R(t) \) and (if it can) \( i_A^L(t) \). The details of how A and B carry out
these operations are what distinguishes one protocol from another. Note that in this
chapter, unlike in [44], we do not assume a two-phase operation, in which a time
slot is divided into two phases, and the two sources take turns to transmit local data
during the first phase, and then take turns again to act (if possible) as a relay during
the second phase. Instead, as in [45] and [46], there is a single “phase” (for Node A,
the first half of the time slot, and for Node B, the second half) in which each source
transmits both its own local data and acts as a relay. This single-phase operation
makes it possible to “combine” the local and relayed data in such a way that a source
is able to exploit its knowledge of what the other source has relayed to improve the
partner-to-partner link.

All four channels realized during time slot \( t \) – the two partner-to-partner channels
and the two partner-to-destination channels – are corrupted by independent, block-
wise Rayleigh fading and complex additive white Gaussian noise (AWGN) with the
same variance \( N_0 \). Perfect channel information (the block fading factor and the noise
variance \( N_0 \)) is available to each receiver, but not to the transmitters. We also assume
that each receiver is capable of detecting decoding failures and never inadvertently
accepts an incorrectly decoded codeword as correct. When LDPC block codes are used to protect the information bits, this failure detection can be effected through the code’s parity-check matrix, and in practice, simulations shows that the probability of undetected errors is negligible and has no discernable effect on performance.

During time slot $t$, when, for instance, Node A is to convey both $i^A_L(t)$ and $i^B_L(t-1)$, there are many ways this can be carried out. What is common to all of the approaches in this chapter is this: Node A encodes $i^A_L(t)$ onto a local codeword $C^A_A(t)$ and encodes $i^B_L(t-1)$ onto a relay codeword $C^A_R(t)$. Node A then combines $C^A_A(t)$ and $C^A_R(t)$ in some way prior to modulation and transmits the combination. What distinguishes one protocol from another is how $C^A_A(t)$ and $C^A_R(t)$ are computed and how they are combined. To clarify the exposition, we make these distinctions:

- After having decoded $i^B_L(t-1)$, how does Node A represent this information in the relay codeword $C^A_R(t)$? We consider two approaches from the literature:
  - The relay codeword is just a copy of Node B’s most recent local codeword, i.e., $C^A_R(t) = C^B_L(t-1)$. We call this a repetition approach.
  - The relay codeword is formed not by repeating $C^B_L(t-1)$ but by computing additional parities for $C^B_L(t-1)$. We call this a coded cooperation approach.

- How does Node A combine the local codeword $C^A_A(t)$ and the relay codeword $C^A_R(t)$ prior to modulation? Again, we consider two approaches:
  - They are combined through simple concatenation, i.e., the bits comprising $C^A_R(t)$ are appended onto the end of $C^A_A(t)$ and the resulting block is passed to the modulator. In this way, each transmitted symbol label is comprised wholly of bits from $C^A_A(t)$ or bits from $C^A_R(t)$, but no transmitted symbol conveys both local and relayed data. We call this approach a time-sharing approach. Note that the term “time-sharing” is used here to describe how each source node uses its time when it is its turn to transmit – first as a
transmitter of local data and then as a relayer of data. Of course, every system in this chapter employs “time sharing” in the sense that the two sources take turns transmitting.

– They are combined via a more complex multiplexer such that each transmitted symbol conveys both local and relayed data. We are in particular interested in carrying out this multiplexing so that the relayed bits in each label identify a particular sparse subset of the QAM/PSK constellation, an approach we call a set partitioning approach.

The primary focus of this chapter is a system employing repetition and set partitioning in its cooperation protocol. However, other approaches are considered for comparison; for example, time-sharing with repetition is used as a reference because it represents the “classic” approach from [39], and the coded cooperation approach is analyzed from an outage perspective. It is worth noting that when the repetition approach is coupled with set partitioning, while the same binary codeword $C^A_L(t) = C^R_R(t)$ is used to represent $i^A_L(t)$ as both local and relayed data, the transmitted signals are different because the modulator treats local data $C^A_L(t)$ and relayed data $C^B_R(t)$ differently. Thus, the resulting gain is not a simple “power gain” as it would be for binary modulation but a more potent “coding gain” [44].

3.3 Cooperation via Set Partitioning and Repetition

Consider a cooperative protocol using set partitioning and repetition (SP-R). Without loss of generality, consider the operations associated with conveying $i^A_L(t)$, the local data generated at Node A during time slot $t$: 1) transmitting at Node A, 2) receiving at Node B, and 3) receiving at Node D.
3.3.1 SP-R: Node A Transmitting

Node A transmits its own local data $i^A_L(t) \in \{0, 1\}^k$ and potentially relays $i^B_L(t-1) \in \{0, 1\}^k$ as well. If $i^B_L(t-1)$ has been decoded successfully, Node A operates in the cooperative transmission mode, illustrated in Figure 3.2:

- $i^A_L(t)$ and $i^B_L(t-1)$ are encoded separately into $n$-bit codewords $C^A_L(t)$ and $C^A_R(t)$, using the same rate-$R = k/n$ LDPC block code. The encoders are systematic.

- After that, a $2^{2m}$-QAM/PSK modulator is used, where $m$ is an integer that divides $n$. To generate the modulator input, Node A takes, in turn, $m$ bits from $C^A_R(t)$ and $m$ bits from $C^A_L(t)$, thereby forming a $2n$-bit sequence $C^A_A(t)$, in which the first $m$ bits are from $C^A_R(t)$, the second $m$ bits are from $C^A_L(t)$, the third $m$ bits are from $C^A_R(t)$, the fourth $m$ bits are from $C^A_L(t)$, etc. In this way, $C^A_A(t)$ is “chopped” into $n/m$ labels; each label contains $2m$ bits, and the $m$ high-order bits in each label are from $C^A_R(t)$, while the $m$ low-order bits are from $C^A_L(t)$. We use a multiplexer $\mathcal{P} \in \{L, R\}^{2n}$ to describe this process, and due to periodicity, only one period is needed for description:

$$\mathcal{P} = [R, R, \ldots, R, L, L, \ldots, L].$$

- The modulator maps each label in $C^A_A(t)$ onto a $2^{2m}$-QAM/PSK symbol to
produce the transmitted frame $s^A(t)$ consisting of $n/m$ QAM/PSK symbols. Moreover, set partition (SP) labeling is used rather than Gray labeling as in conventional bit-interleaved coded modulation systems. Figure 3.3(a) and Figure 3.3(b) show the labeling for 16-QAM ($m = 2$) and 64-QAM ($m = 3$), respectively. The $m$ high-order bits partition the constellation into $2^m$ sparse subsets, and in each subset there are $2^m$ symbols specified by the $m$ low-order bits. Thus, the $m$ relayed bits first select a sparse subset, and then the $m$ local bits select a particular symbol from this subset.

On the other hand, if $i^B_L(t - 1)$ has not been decoded successfully, then Node A operates in the non-cooperative transmission mode, and only $i^A_L(t)$ is conveyed. To keep the spectral efficiency constant, $i^A_L(t)$ is systematically encoded onto a $2n$-bit codeword $C^A_L(t)$ with a rate $R/2 = k/2n$ LDPC block code. We refer to the rate-$R/2$ code as the “low-rate” code in contrast to the “high-rate” code used in the cooperative mode. Then $C^A(t) = C^A_L(t)$, and the modulator now adopts Gray labeling.

In both the cooperative and non-cooperative modes, since a $k$-bit packet of local data is always transmitted via $n/m$ QAM/PSK symbols, the resulting spectral efficiency is

$$\eta = k/(n/m) = Rm \text{ bits per symbol.}$$

This encoding-multiplexing-modulating procedure can be generalized to $2^{2m+1}$-QAM/PSK modulation using four-dimensional signaling. The necessity of four-dimensional signaling is obvious: the $(2m + 1)$ bits forming a single label cannot be shared equally by local and relayed data. As a result, we design the multiplexer so that the odd-numbered two-dimensional symbols convey $(m + 1)$ relayed bits and $m$ local bits while the even-numbered symbols convey $m$ relayed bits and $(m + 1)$
Figure 3.3. The SP labeling for 16-QAM and 64-QAM modulation: those solid signal points have the same high-order half of the labeling.
local bits, an approach represented by $\mathcal{P}$:

$$\mathcal{P} = [R, \ldots, R; L, \ldots, L; R, \ldots, R, L, \ldots, L]_{m+1}.$$ 

Now the constraint is that $(2m + 1)$ must divide $2n$.

Note: Before recovering $i^A_L(t)$, the receiver – at Node B or D – must know whether the associated signal was transmitted in the cooperative mode or non-cooperative mode. We assume that this side information is always available to the receiver. In practice, this can be achieved in two possible ways, as described in [46]: 1) a robust flag bit can be transmitted simultaneously with the frame, or 2) the receiver can attempt to decode using all of the transmission modes and use the error-detecting capability of the channel codes to see which decoding mode yields a reliable packet – or, if none do, indicates a decoding failure.

3.3.2 SP-R: Node B Receiving

It is the job of Node B to recover $i^A_L(t)$ from the received frame. If the frame was transmitted in the non-cooperative mode, a conventional decoder for the rate $\mathcal{R}/2$ block code may be used. Conversely, if the frame was transmitted in the cooperative mode, then Node B, with its knowledge of $i^B_L(t - 1)$, knows the values of $C^A_R(t)$ that were used to select the $m$ high-order bits of each transmitted symbol. Thus, Node B knows the sequence of sparse subsets from which the transmitted symbols were drawn and can use that information to provide a better estimate of $i^A_L(t)$. For LDPC block codes, this means that the belief-propagation (BP) decoding algorithm can be initialized with more reliable soft log-likelihood ratio (LLR) values for each bit. Figure 3.4 illustrates the above receiver structure.
3.3.3 SP-R: Node D Receiving

Node D estimates $i^A_L(t)$ based on block-faded, noisy received versions of $s^A(t)$ and (potentially) $s^B(t)$, i.e., the frame(s) transmitted by Node A and (potentially) Node B during time slot $t$. Let $y^A(t)$ and $y^B(t)$ denote the received frames corresponding to $s^A(t)$ and $s^B(t)$, respectively. Then, depending on which of those two frames contain (or do not contain) relayed data, there are four possible “states” in which the Node D receiver will operate:

1. State $S_0$: $s^A(t)$ contains $i^A_L(t)$ but not $i^B_L(t - 1)$, and $s^B(t)$ contains $i^B_L(t)$ but not $i^A_L(t)$, i.e., neither Node A or B cooperated during time slot $t$. 

Figure 3.4. Receiver in the SP-R approach at Node B: decoder initialization is more reliable when the SP labeling is adopted.
TABLE 3.1

FOUR POSSIBLE STATES: NODE D DECODING $i_A^l(t)$

<table>
<thead>
<tr>
<th>State</th>
<th>Codeword</th>
<th>Rate</th>
<th>Frame</th>
<th>Diversity Achieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$C_A^l(t)$</td>
<td>$k/(2n)$</td>
<td>$y_A^l(t)$</td>
<td>No</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$C_A^l(t)$</td>
<td>$k/(2n)$</td>
<td>$y_A^l(t)$</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>$C_B^R(t)$</td>
<td>$k/n$</td>
<td>$y_B^l(t)$</td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td>$C_A^l(t)$</td>
<td>$k/(n)$</td>
<td>$y_A^l(t)$</td>
<td>No</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$C_A^l(t)$</td>
<td>$k/(n)$</td>
<td>$y_A^l(t)$</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>$C_B^R(t)$</td>
<td>$k/n$</td>
<td>$y_B^l(t)$</td>
<td></td>
</tr>
</tbody>
</table>

2. State $S_1$: $s_A^l(t)$ contains $i_A^l(t)$ but not $i_L^B(t-1)$, and $s_B^l(t)$ contains both $i_L^B(t)$ and $i_A^l(t)$, i.e., Node A did not cooperate, but Node B cooperated.

3. State $S_2$: $s_A^l(t)$ contains both $i_A^l(t)$ and $i_L^B(t-1)$, and $s_B^l(t)$ contains $i_L^B(t)$ but not $i_A^l(t)$, i.e., Node A cooperated, but Node B did not.

4. State $S_3$: $s_A^l(t)$ contains both $i_A^l(t)$ and $i_L^B(t-1)$, and $s_B^l(t)$ contains both $i_L^B(t)$ and $i_A^l(t)$, i.e., both Node A and B cooperated.

These decoding states of Node D for $i_A^l(t)$ are also summarized in Table 3.1. Obviously, $y_A^l(t)$ is always used for decoding, since the codeword $C_A^l(t)$ in it always contains $i_A^l(t)$. And whenever $y_B^l(t)$ contains $C_B^R(t)$ which was generated based on $i_A^l(t)$, cooperative diversity is achieved, thus benefiting the decoding process.

The specific operation of the decoder in each state is code-dependent. In this section we will still assume that LDPC block codes are used. Let $\text{LLR}_{\text{ini}}(\cdot)$ denote initial LLR values in the BP decoding algorithm.
In state $S_0$, as summarized in Table 3.1, only one version of $i^A_L(t)$ is received, so no
diversity is available. A conventional BP decoder for the rate-$R/2 = k/(2n)$ LDPC
block code is used to recover $i^A_L(t)$ from $y^A(t)$.

In state $S_2$, again, no diversity is available. However, the SP labeling is adopted
in $y^A(t)$, so if $i^B_L(t-1)$ has been decoded successfully in the last decoding round, i.e.,
if $C^A_R(t)$ is already known to Node D, then the decoder operates based on sparse sub-
constellations. Otherwise, the decoder operates based on the original constellation.

In state $S_1$, $i^A_L(t)$ is received twice at two different rates: first via $C^A_L(t)$ in $y^A(t)$
at rate $R/2$, and then via $C^B_R(t)$ in $y^B(t)$ at rate $R$. The decoding process of state
$S_1$ is shown in Figure 3.5. Pertinent points:

- Cooperative diversity is achieved. Moreover, because both codes are system-
atic, the information parts of both $C^A_L(t)$ and $C^B_R(t)$ are identically equal to
$i^A_L(t)$. Therefore, the associated soft values can be added to effect maximal
ratio combining (MRC) and thereby improve the decoder initialization when
decoding either codeword.

- It may be possible to improve the initial soft values for $i^A_L(t)$ even more by
first attempting to decode $C^B_L(t)$ based on $y^B(t)$. If successful, this identifies
the low-order bits in the labels of the transmitted symbols making up $s^B(t)$,
i.e., it identifies the sequence of sub-constellations from which the transmitted
symbols are drawn. Note that these sub-constellations are not the sparse sub-
constellations defined by the high-order bits; for instance, in Figure 3.3(a),
knowing the low-order bits in a 16-QAM label identifies the quadrant containing
the symbol. Both analysis and simulation show that soft values for $i^A_L(t)$ based
on these sub-constellations are more reliable than soft values based on the larger,
original constellation. Therefore, the first thing the decoder in Figure 3.5 does
is attempt to decode $C^B_L(t)$ and, if successful, use that success to improve the
Figure 3.5. The SP-R approach: Node D decodes $i_L^A(t)$ in state $S_1$. 
soft values associated with $i^A_L(t)$:

\[
\text{LLR}_{\text{ini}} (i^A_L(t)) = \text{LLR}_{\text{ini}} (i^A_L(t)|y^A(t)) + \begin{cases} \\
\text{LLR}_{\text{ini}} (i^A_L(t)|y^B(t), C^B_L(t)) & \text{if } C^B_L(t) \text{ is known} \\
\text{LLR}_{\text{ini}} (i^A_L(t)|y^B(t)) & \text{otherwise}
\end{cases}
\]

- Given these improved initial soft values associated with $i^A_L(t)$, Node D now has two opportunities to recover $i^A_L(t)$ from two different codewords. The receiver attempts to decode $C^A_L(t)$ first because it has a lower rate and is more likely to be successfully decoded; if the decoding fails, then $C^B_R(t)$ is decoded.

State $S_3$ is the other state in which cooperative diversity is available. The data $i^A_L(t)$ has been encoded into two identical codewords $C^A_L(t)$ and $C^B_R(t)$ which are transmitted as the low-order bits of the labels in $s^A(t)$ and the high-order bits of the labels in $s^B(t)$, respectively. The decoder initialization in state $S_3$ is similar to that in state $S_2$ in that, if $i^B_L(t - 1)$ has been decoded successfully in the last round, then the soft values derived from $y^A(t)$ can be based on sparse subsets and thus improved. Similarly, the decoder initialization in state $S_3$ can employ the approach taken in state $S_1$ by first attempting to decode $C^B_R(t)$ based on $y^B(t)$ and using those results (if successful) to improve the soft values derived from $y^B(t)$. The operation of the decoder in State $S_3$ can be summarized as follows:

\[
\text{LLR}_{\text{ini}}(C^A_L(t)) = \\
\begin{cases} \\
\text{LLR}_{\text{ini}}(C^A_L(t)|y^A(t), C^A_R(t)) & \text{if } C^A_R(t) \text{ is already known}; \\
\text{LLR}_{\text{ini}}(C^A_L(t)|y^A(t)) & \text{otherwise};
\end{cases}
\]

\[
+ \\
\begin{cases} \\
\text{LLR}_{\text{ini}}(C^B_R(t)|y^B(t), C^B_L(t)) & \text{if } C^B_L(t) \text{ is already known}; \\
\text{LLR}_{\text{ini}}(C^B_R(t)|y^B(t)) & \text{otherwise}.
\end{cases}
\]

82
Note that since $C^A_L(t)$ and $C^B_R(t)$ are the same codewords, bit-wise MRC can improve the initial soft values for every bit in the whole codeword. Then, a single attempt to decode that codeword is made.

3.3.4 Simulation: Frame Error Rate at Destination

Performance of the SP-R approach is evaluated via simulation using frame error rate (FER) at Node D as the metric of interest. Here, the FER indicates the fraction of $k$-bit data packets ($i^A_L(t)$ and $i^B_L(t)$) that cannot be recovered at Node D.

Figure 3.6 displays the FER when 8-PSK modulation is used with $k = 1000$ bit data packets and a spectral efficiency of $\eta = 1$ bit/symbol. Also included for comparison are the results for time-sharing repetition-based cooperation (TS-R) in which the relayed codeword $C^A_R(t) = C^B_L(t-1)$ and the local codeword $C^A_L(t)$ are
simply concatenated prior to modulation, and for a non-cooperative system in which the source nodes never cooperate but instead consistently use the low-rate $R = k/(2n)$ LDPC block code with Gray labeling. The new approach SP-R offers a coding gain of about 3.3 dB over TS-R at a FER of $10^{-2}$.

Similarly, Figure 3.7 displays the results for 64-QAM when $k = 1500$-bit data packets are used with a spectral efficiency of $\eta = 1.5$ bits/symbol, indicating a performance gain of 2.7 dB for SP-R relative to TS-R. Simulation results for other modulation schemes and spectral efficiency values can be found in [50].
3.4 Set-Partitioning Coded Cooperation (SP-CC): Achieving Different Cooperation Levels

The repetition-based approaches described in the last section – the conventional TS-R and the new SP-R – always use the same binary codeword to convey a given data packet, whether that data packet represents local information or relayed information. In contrast, a bandwidth-efficient approach based on “coded cooperation” would relay not the same binary codeword (appropriately modulated) but instead additional parities. This approach offers the possibility of changing the system’s cooperation level, a parameter \( \beta \) defined in [44] as the fraction of transmitted bits that a node dedicates to relaying when operating in the cooperative transmission mode:

\[
\beta = \frac{\text{The number of transmitted relayed bits}}{\text{The total number of transmitted bits}}.
\]

For repetition-based cooperation, the cooperation level is necessarily \( \beta = 0.5 \). In this section, we briefly demonstrate two methods for achieving other values of \( \beta \) for QAM/PSK modulation.

3.4.1 A Multiplexer-Based Approach

Assume the modulation is \( 2^M \)-ary, where \( M = m_L + m_R \) for positive integers \( m_L \) and \( m_R \), and \( \beta = m_R/M \). Then, in cooperative mode, each modulation symbol will have a binary label with \( m_L \) low-order bits selected by the (encoded) local data and \( m_R \) high-order bits selected by the (encoded) relayed data. To achieve this, the \( k \) local bits are encoded with a rate-\( R_L \) encoder while the \( k \) relayed bits are encoded with a rate-\( R_R \) encoder, where \( R_L \) and \( R_R \) satisfy \( \beta = \frac{R_L}{R_L + R_R} \). The net result is a local codeword that is \( n_L = k/R_L \) bits long and a relayed codeword that is \( n_R = k/R_R \).
bits long; they are combined into an \((n_L + n_R)\)-bit block via a multiplexer

\[
P = [R, R, \ldots, R, L, L, \ldots, L].
\]

Thus, the multiplexer produces a sequence of \(n_L/m_L = n_R/m_R\) labels, each \(M\) bits long. The \(2^M\)-ary constellation is partitioned into \(2^{m_R}\) sparse subsets, each containing \(2^{m_L}\) symbols, and under the mapper the (encoded) relayed bits select a subset while the (encoded) local bits select a symbol from the subset. The spectral efficiency of the code is \(\eta = R_L m_L = R_R m_R\) bits/symbol.

As an example, consider a 16-QAM system in which local data is encoded at a rate \(R_L = 1/4\) and relayed data is encoded at a rate \(R_R = 3/4\). Then a \(k = 1200\)-bit local packet produces an \(n_L = 4800\)-bit local codeword and a \(k = 1200\)-bit relayed packet produces an \(n_R = 1600\)-bit relayed codeword. The resulting 6400 bits are transmitted over 1600 16-QAM symbols, with each symbol label formed by taking a single \((m_R = 1)\) high-order bit from the relayed codeword and the three \((m_L = 3)\) low-order bits from the local codeword. Moreover, the mapping is done so that the high-order bit in each label partitions the 16-QAM constellation into two sparse subset of eight symbols each. The resulting spectral efficiency is \(\eta = 0.75\) bits/symbol.

### 3.4.2 A Mixing Approach

An obvious disadvantage of the multiplexer-based approach is its coarse granularity in \(\beta\). With 16-QAM, the only possible values of \(\beta\) using the multiplexer-based scheme are \(\beta = 1/4, 1/2, \text{ and } 3/4\). However, by “mixing” two different kinds of QAM symbols – some with labels formed only from local data and others with labels formed from both local and relayed data – a wider variety of values of \(\beta\) are possible. For instance, consider a cooperative scheme in which local data is encoded at a rate \(R_L = 1/2\) and relayed data at a rate \(R_R = 3/4\). If each data packet is \(k = 1200\)
bits, then the resulting $n_L = 2400$-bit local codeword and the $n_R = 1600$-bit relayed codeword would be used to modulate 1000 16-QAM symbols, for instance, with 800 QAM symbols labeled with two bits of relayed data and two bits of local data, and the other 200 QAM symbols labeled only by local data. Such a scheme would have a cooperation level of $\beta = 0.4$ and a spectral efficiency of $\eta = mR_LR_R/(R_L + R_R) = 1.2$ bits/symbol, where $2^m = 16$ is the constellation size. Essentially, the time-sharing structure is incorporated, since the “local-relay” based symbols and the “local-only” based symbols are concatenated into a transmitted frame.

3.5 Outage Analysis of Cooperation Based on Set Partitioning

This section analyzes the outage behavior of cooperation based on set partitioning; in essence, this analysis assumes that capacity-achieving codes are available for each link and determines the probability that the randomly-chosen channel realizations are sufficient to allow those codes to deliver data to the destination.

For a given modulation choice and a given desired spectral efficiency $\eta$, the approach described above dictates a particular rate on each of the four links making up the cooperative system. If the instantaneous signal-to-noise ratio (SNR) on a given link is insufficient to provide the necessary capacity, then a link outage occurs on the receiving end of that link. A system outage occurs when link outages combine to prevent the successful delivery of a packet to the destination. For simplicity of exposition, 16-QAM is used in this section, but the analysis can easily be generalized to other QAM/PSK modulation schemes.

We start with the outage analysis of the set-partitioning repetition (SP-R) approach.
3.5.1 SP-R Outage: At the Partner Nodes

Link outages at each partner determine the subsequent transmission modes (cooperative or non-cooperative) which in turn determine the decoding state at the destination. Outage events can occur under two circumstances:

- If the received frame was transmitted in the non-cooperative mode, then the outage probability is given by

  \[ P \left\{ C_{16QAM}(\gamma_{S \rightarrow R}) < \eta \right\} , \]

  where \( \eta \) is the desired spectral efficiency and \( C_{16QAM}(\gamma_{S \rightarrow R}) \) is the capacity of Gray-coded 16-QAM modulation with a (randomly generated) source-to-relay SNR of \( \gamma_{S \rightarrow R} \).

- If the received frame was transmitted in the cooperative mode using SP labeling, the outage probability is given by

  \[ P \left\{ C_{\text{FH}}(\gamma_{S \rightarrow R}) < \eta \right\} . \]

  Here, “FH” means “fixed high-order,” which refers to the fact that, with knowledge of the two high-order bits in the label, each symbol is restricted to one of the sparse subsets of the original constellation. There are (for 16-QAM) four such subsets, each with four symbols; however, they all have the same capacity as a function of SNR, designated \( C_{\text{FH}}(\gamma_{S \rightarrow R}) \).

3.5.2 SP-R Outage: At the Destination Node

An outage at the destination indicates the inability of the system to reliably deliver an information packet to Node D. This depends on the state in which the Node D receiver is operating, which, as noted above, further depends on the outage
events that previously occurred at the partner nodes. Consider the operations of Node D in decoding $i_L^A(t)$:

3.5.2.1 State $S_0$

In state $S_0$, the data $i_L^A(t)$ was transmitted only from Node A, with no diversity. Moreover, the frame to be decoded, $s^A(t)$, was generated in the non-cooperative mode, so the low-rate code was used with the complete, Gray-coded 16-QAM constellation. Thus, the outage probability is

\[
P \{ C^{16\text{QAM}}(\gamma_{S \rightarrow D}) < \eta \},
\]

where $\gamma_{S \rightarrow D}$ is the (randomly generated) SNR on the source-to-destination (A-to-D) link.

3.5.2.2 State $S_1$

Recall that cooperative diversity is available in state $S_1$. The data $i_L^A(t)$ was first encoded into $C_L^A(t)$ with the low-rate code and transmitted by Node A as the only contents of $s^A(t)$; it was then encoded onto $C_B^R(t)$ with the high-rate code and transmitted by Node B in the two high-order bits of the labels in $s^B(t)$.

To analyze the probability of an outage under these circumstances, an optimistic (“genie-aided”) simplification is made – that Node D has available to it the two low order bits of the symbol labels in $s^B(t)$. Looking at Figure 3.3(a), this means that Node D knows a priori in which quadrant each symbol in $s^B(t)$ lies. Making this assumption yields a lower bound on the outage probability and makes the required analysis computationally feasible. Moreover, this assumption is supported by the decoding algorithm in Section 3.3.3, wherein an initial attempt is made to determine the low-order bits in advance of the high-order bits by decoding $C_L^B(t)$ first. Finally,
simulation results indicate that this assumption provides a tight lower bound.

With this simplification in mind, three different kinds of information about \( \hat{i}^A_L(t) \) are available to Node D:

- Two noisy versions of \( \hat{i}^A_L(t) \), appearing as the systematic parts of \( C^A_L(t) \) and \( C^B_R(t) \). Direct transmission (from A to D) is associated with Gray-coded 16-QAM modulation, while with the “genie” assumption, relayed transmission (from B to D) is associated with a sequence of known sub-constellations, each comprising one quadrant of 16-QAM.

- One noisy version of the parities generated by the non-cooperative encoder, transmitted using Gray-coded 16-QAM from Node A to Node D.

- One noisy version of the parities generated by the cooperative encoder, transmitted using the quadrant sub-constellations from Node B to Node D.

The system is in outage if the two links do not provide sufficient capacity to support the decoding of \( \hat{i}^A_L(t) \). Let \( \mathcal{R} = \eta/m \) (the constellation size is \( 2^{2m} \), and \( m = 2 \) here) denote the rate of the code used in the cooperative mode, and let \( \mathcal{R}_{NC} = \mathcal{R}/2 \) be the rate used in the non-cooperative mode. Then, the outage probability is given by

\[
P\{\mathcal{R}_{NC}C^{16\text{QAM}}_{\text{FL, FL}}(\gamma_S \rightarrow D, \gamma_R \rightarrow D) + (1 - \mathcal{R}_{NC})C^{16\text{QAM}}(\gamma_S \rightarrow D) + (1 - \mathcal{R})C^{\text{FL}}(\gamma_R \rightarrow D) < \eta}\.
\]

Here, “FL” means “fixed low-order,” which refers to the sub-constellations obtained when the genie-provided \( m \) low-order bits in each label are fixed. The capacity term \( C^{16\text{QAM}}_{\text{FL, FL}}(\gamma_S \rightarrow D, \gamma_R \rightarrow D) \) corresponds to the channel over which the first kind of information referenced above is transmitted. The MRC-enhanced transmission of \( \hat{i}^A_L(t) \) can be viewed as signaling over two parallel channels: one channel at an SNR...
of $\gamma_{S \rightarrow D}$ using Gray-coded 16-QAM modulation, and the other channel at an SNR of $\gamma_{R \rightarrow D}$ using the two high-order bits in the labels of two consecutive 16-QAM symbols.

3.5.2.3 State $S_2$

In state $S_2$, no diversity is available. Only the noisy version of $s^A(t)$ is used for decoding and it was generated in the cooperative mode, i.e., $i^A_L(t)$ was encoded onto $C^A_L(t)$ with the high-rate code. In this case, the outage probability depends on whether the relayed data $C^A_R(t)$ is already known to the receiver. As described in Section 3.3.3, if Node D has already recovered $i^R_L(t-1)$, then $i^A_L(t)$ will be decoded based on the sparse sub-constellations with side knowledge $C^A_R(t)$; otherwise, $i^A_L(t)$ will be decoded based on the whole constellation.

Thus, if $C^A_R(t)$ is known, the outage probability is defined as

$$P\{C^{FH}(\gamma_{S \rightarrow D}) < \eta\}.$$  

On the other hand, if $C^A_R(t)$ is not known, then the receiver attempts to recover $i^A_L(t)$ from the high-rate $C^A_L(t)$ transmitted in the two low-order bits of consecutive 16-QAM symbols without knowledge of the two high-order bits; for example, “00” is represented by a 4-ary sub-constellation rather than by a single symbol. Let “UH” denote “unknown high-order” to designate this modulation scheme, then the outage probability is defined as

$$P\{C^{UH}(\gamma_{S \rightarrow D}) < \eta\}.$$
3.5.2.4 State $\mathcal{S}_3$

In state $\mathcal{S}_3$, cooperative diversity is available. Both $s^A(t)$ and $s^B(t)$ are used for decoding, and they are both in the cooperative mode. Node D observes two noisy versions of the same codeword $C^A_L(t) = C^B_R(t)$; the former is transmitted via the low-order bits of the symbols on the source-to-destination link, while the latter is transmitted via the high-order bits of the symbols on the relay-to-destination link.

Following the analysis of state $\mathcal{S}_1$ and $\mathcal{S}_2$,

- $C^B_L(t)$ in $s^B(t)$ is genie-provided, i.e., the receiver is assumed to know the lower-order bits of each symbol in $s^B(t)$, and

- The availability of the side information $C^A_R(t)$ in $s^A(t)$ depends on the decoding result of $i^B_L(t - 1)$ from the last decoding round.

With regard to the second point, if $i^B_L(t - 1)$ has already been decoded successfully, i.e., if $C^A_R(t)$ is known, then the outage probability is given by

$$P \{ C_{FL}^{FH}(\gamma_{S \rightarrow D}, \gamma_{R \rightarrow D}) < \eta \}.$$ 

On the other hand, if $i^B_L(t - 1)$ has not been decoded successfully, then the outage probability is

$$P \{ C_{FL}^{UH}(\gamma_{S \rightarrow D}, \gamma_{R \rightarrow D}) < \eta \}.$$ 

Here again, the process of exploiting cooperative diversity via MRC is viewed as transmission over parallel channels. As before, we use the superscripts (for the first frame) “FH” and “UH” to characterize the status of the high-order bits (either fixed or unknown), and use the subscript (for the second frame) “FL” to indicate that the low-order bits are fixed.
3.5.3 Outage Analysis for Other Systems

To provide a comparison between the new set-partitioning-repetition (SP-R) approach and the more conventional time-sharing-repetition (TS-R) approach as well as the set-partitioning-coded-cooperation (SP-CC) approach, similar outage analyses can be applied.

3.5.3.1 Time-Sharing Repetition

Using the TS-R approach, the two codewords $C_A^A(t)$ and $C_R^A(t)$ are simply concatenated before 16-QAM modulation at Node A, so the fact that Node B knows $C_R^A(t)$ does not help Node B in decoding $C_A^A(t)$. Thus, in the cooperative mode, the outage probability at the partner is given by

$$P \{ C_{16\text{QAM}}(\gamma_S \rightarrow R) < 2\eta \}.$$

The outage probability in the non-cooperative mode is the same as that in the SP-R approach.

Moreover, the resulting system outage probabilities at the destination can be shown to be as follows:

- State $S_0$:

  $$P \{ C_{16\text{QAM}}(\gamma_S \rightarrow D) < \eta \},$$

- State $S_1$:

  $$P \{ R_{NC}C_{16\text{QAM}}(\gamma_S \rightarrow D, \gamma_R \rightarrow D) + (1 - R_{NC})C_{16\text{QAM}}(\gamma_S \rightarrow D) + \frac{1 - R_{NC}}{2}C_{16\text{QAM}}(\gamma_R \rightarrow D) < \eta \},$$
• State $S_2$:

$$P \left\{ C_{16QAM}^{16QAM} (\gamma_S \rightarrow D) < 2\eta \right\},$$

and

• State $S_3$:

$$P \left\{ C_{16QAM}^{16QAM} (\gamma_S \rightarrow D, \gamma_R \rightarrow D) < 2\eta \right\}.$$

### 3.5.3.2 Set-Partitioning Coded Cooperation

As for the SP-CC approach, let $\alpha = 1 - \beta$, i.e.,

$$\alpha = \frac{\text{Number of transmitted bits representing local information}}{\text{Total number of transmitted bits}}.$$

Consider the case when the mixing approach is employed and $\beta$ varies is between 0 to 0.5. Then, in the cooperative mode, the outage probability is given by

$$P \left\{ (\alpha - \beta)C_{16QAM}^{16QAM} (\gamma_S \rightarrow R) + 2\beta C_{FH}^{16QAM} (\gamma_S \rightarrow R) < \eta \right\}.$$

At the destination node, the outage probabilities are given by:

• State $S_0$:

$$P \left\{ C_{16QAM}^{16QAM} (\gamma_S \rightarrow D) < \eta \right\};$$

• State $S_1$ (genie-derived lower bound):

$$P \left\{ \alpha C_{16QAM}^{16QAM} (\gamma_S \rightarrow D) + \beta C_{FL, FL}^{16QAM} (\gamma_S \rightarrow D, \gamma_R \rightarrow D) < \eta \right\};$$
• State $S_2$: depending on whether the relayed codeword in the first frame is already known,

\[ P\{ (\alpha - \beta)C^{16QAM}(\gamma_{S \rightarrow D}) + 2\beta C^{FH}(\gamma_{S \rightarrow D}) < \eta \} , \]

or

\[ P\{ (\alpha - \beta)C^{16QAM}(\gamma_{S \rightarrow D}) + 2\beta C^{UH}(\gamma_{S \rightarrow D}) < \eta \} , \]

and

• State $S_3$ (genie-derived lower bound): again, depending on whether the relayed codeword in the first frame is already known,

\[ P\{ (\alpha - \beta)C^{16QAM}(\gamma_{S \rightarrow D}) + 2\beta C^{FH}(\gamma_{S \rightarrow D}) + 2\beta C^{FL}(\gamma_{R \rightarrow D}) < \eta \} , \]

or

\[ P\{ (\alpha - \beta)C^{16QAM}(\gamma_{S \rightarrow D}) + 2\beta C^{UH}(\gamma_{S \rightarrow D}) + 2\beta C^{FL}(\gamma_{R \rightarrow D}) < \eta \} . \]

3.5.4 Monte Carlo Simulation

This section uses simulation to evaluate the outage behavior of the various cooperation schemes. This is done by pseudo-randomly generating a sequence of channel realizations (i.e., link SNR values) and using the results of the last section to evaluate when outages occur at the partner and destination nodes. The frequency with which outages occur at Node D provides an estimate of the system outage probability.
3.5.4.1 Comparing Two Repetition-Based Schemes: SP-R versus TS-R

Figure 3.8 shows the outage probabilities of the newly-proposed set-partitioning repetition (SP-R) approach for 16-QAM with spectral efficiency $\eta$ taking values of 0.4, 0.8 and 1.2 bits/symbol; also included for comparison are the analogous results for the conventional time-sharing repetition (TS-R) approach. It is observed that the schemes based on set partitioning enjoy a gain that ranges from 1.1 dB (for $\eta = 0.4$ bits/symbol) to 2.8 dB (for $\eta = 1.2$ bits/symbol) over the corresponding schemes based on time sharing for an outage probability of $10^{-2}$. The dependence on $\eta$ is more clearly discerned in Figure 3.9, where the average SNR required to obtain an outage probability of $10^{-2}$ is plotted as a function of $\eta$. The gap between the two curves – representing the gain of SP-R over TS-R – increases with $\eta$, from less that 1.0 dB for $\eta = 0.2$ to 3.5 dB for $\eta = 1.8$. 

96
Spectral Efficiency $\eta$ (bits/symbol)

Average $E_b/N_0$ (dB)

Figure 3.9. Performance gain of SP-R over TS-R: $\eta$ vs. average $E_b/N_0$ (dB) when the outage probability is fixed at $10^{-2}$; 16-QAM.

Figure 3.10 compares the outage probabilities with the frame error rates obtained using LDPC block codes with 16-QAM modulation operating at a spectral efficiency of $\eta = 1.0$ bit/symbol. Information packets containing $k = 1000$ bits were used with rate-$1/2$ LDPC block codes to provide a comparison between the frame error rates (FER) observed via codes of practical length with the outage probabilities computed previously. It is seen that the gaps are typically less than 1.0 dB.

3.5.4.2 Comparison: TS-R, TS-CC, SP-R and SP-CC

Figure 3.11 shows the outage performance comparison of the TS-R, TS-CC, SP-R and SP-CC approaches for the case $\eta = 1.0$ bit/symbol. Simulations conclude that for TS-CC, the nearly-optimal choice of cooperation level is $\beta = 0.3$, while for SP-CC, the nearly-optimal choice should be $\beta = 0.5$. 

97
3.6 Two Variations on the Set-Partitioning Repetition Approach

In this section, we briefly present two variations to the set-partitioning repetition (SP-R) approach to cooperation developed in this chapter.

3.6.1 Turbo-Decoding Structure

Consider the operation of Node D decoding $i_L^A(t)$ in an SP-R system. Analysis and simulation show that at medium-to-high SNR, state $S_3$ is the dominant state at Node D, i.e., with high probability, both frame $s^A(t)$ and frame $s^B(t)$ are generated in the cooperative mode. As stated above, in this case, $i_R^B(t) = i_L^A(t)$ and $C_R^B(t) = C_L^A(t)$, and maximal ratio combining (MRC) can be carried out on the whole codeword, i.e., the belief propagation decoder can be initialized with MRC-enhanced soft information.
for every bit in the codeword.

An alternative approach is as follows. Rather than setting \( i_R^B(t) = i_L^A(t) \), set \( i_R^B(t) = \pi(i_L^A(t)) \) where \( \pi(\cdot) \) is a pseudo-random interleaver; as a result, \( C_R^B(t) \) is now different from \( C_L^A(t) \). The disadvantage of this variation is that in state \( S_3 \), only the common systematic part can be enhanced via MRC. However, the advantage is that now \( C_L^A(t) \) and \( C_R^B(t) \) can be jointly decoded: at the end of each iteration of the belief-propagation decoder for \( C_L^A(t) \), extrinsic soft information of \( i_L^A(t) \) can be extracted and interleaved and used as a-priori\(^2\) soft information about \( i_R^B(t) \) for the decoder for \( C_R^B(t) \). A turbo decoding structure is in effect created, and we refer to this as the turbo-SP-R approach. And obviously, in state \( S_1 \) a similar decoding

\(^2\)As defined in [52], “the a-priori information about a bit is information known before decoding starts, from a source other than the received sequence or the code constraints.”
process can be applied.

The turbo-SP-R approach can be viewed as a “transition” from SP-R to SP-CC. When SP-R is employed, the relayed bits are all repeated versions of the bits transmitted by the originating partner, while when SP-CC is used, the relayed bits are all newly generated. However, with turbo-SP-R, some of the relayed bits are repeated (i.e., the systematic bits), while some are newly generated (i.e., the parities).

Figure 3.12 compares the FER performances of turbo-SP-R and SP-R at Node D with 16-QAM modulation and $\eta = 0.5$ bits/symbol; each information packet contain $k = 500$ bits. It is observed that at FER= $10^{-2}$, turbo-SP-R provides a coding gain of about 0.75 dB, compared to SP-R. These and other simulation results indicate that for low-to-moderate values of the spectral efficiency $\eta$, the turbo-SP-R approach yields some benefit; however, as $\eta$ increases, this benefit vanishes.
3.6.2 Frame Interleaving

It is well known that one of the appealing features of spatial diversity is that it can be combined with other forms of diversity. In this section, to further improve system performance, temporal diversity is incorporated into cooperative diversity through frame interleaving.

As described and analyzed above, in a SP-R system, each data packet is encoded and transmitted twice – first as local data and then as relayed data – so cooperative diversity is achieved. However, each codeword “sees” only a single channel realization, and thus deep block fading may render an entire codeword useless to the receiver. The following variation, inspired by the conventional bit interleaved coded modulation approach, splits codewords over multiple channel realizations. Its logical extreme would be a system in which every bit in a given codeword experiences independent and identically distributed channel realizations, just as the model considered in [3].

Node A transmission of $i^A_L(t)$ is modified as follows:

- All the operations shown in Figure 3.2 remain the same.

- A frame interleaver is inserted between the modulator output $s^A(t)$ and the channel: $s^A(t)$ is divided into $d$ successive “atomic frames” with the same length, namely $s^A(t, 1), s^A(t, 2), \ldots, s^A(t, d)$; $s^A(t, i), i = 1, 2, \ldots, d$ will wait in the queue and be transmitted in time slot $(t + i - 1)$, so each atomic frame from $s^A(t)$ experiences a different (independent) channel realization. Moreover, rather than transmitting $s^A(t)$ during time slot $t$, Node A actually transmits the concatenation of $d$ atomic frames from $d$ different codewords:

  \[
  [s^A(t - d + 1, d), s^A(t - d + 2, d - 1), \ldots, s^A(t, 1)].
  \]

- We refer to $d$ as the “frame division factor”, and this variation as the $d$-SP-R
approach. The original SP-R approach is obviously the special case when $d = 1$.

The decoding latency at the partner is approximately $d$ time slots. At moderate- to-high SNR, cooperation is the dominant mode, so with high probability each codeword is transmitted to the destination over $2d$ i.i.d. channel realizations – the first $d$ realizations as local data and the rest as relayed data – and the decoding latency at the destination is about $2d$ time slots.

Figure 3.13 shows the simulation results for the $d$-SP-R approach when $d$ varies from 1 (the SP-R approach) to 2, 4 and 8. As expected, the diversity order increases as $d$ increases, and the performance gain is, for example, approximately 3.0 dB from $d = 2$ to $d = 4$ at a FER of $10^{-2}$. The outage behavior of the symbol-wise fully-interleaved case (i.e., each atomic frame consists of only one symbol, so each codeword is transmitted over a channel with independent fading for every QAM symbol) is also evaluated. This analysis is carried out with three optimistic assumptions:

- The partner-to-partner links are perfect, so Node D always operates in state $S_3$ and spatial diversity is always available.

- When the “target” data is transmitted on the source-to-destination channel as the local data using the two low-order bits in each symbol, assume that the two high-order bits in each symbol are already known, i.e., “fixed high-order” (FH) as in Section 3.5.2.

- When the data is transmitted on the relay-to-destination channel as the relayed data using the two high-order bits in each symbol, assume that the two low-order bits in each symbol are already known, i.e., “fixed low-order” (FL).

Under the above assumptions, the outage probability is

$$P \left\{ \sum_{i=1}^{N} \frac{1}{N} C_{\text{FH}}^{\text{FL}} \left( \gamma_S \rightarrow D(i), \gamma_R \rightarrow D(i) \right) < \eta \right\},$$

102
Figure 3.13. The $d$-SP-R approach, FER comparison at Node D: 16-QAM, $\eta = 1.0$ bit/symbol, $k = 1000$.

where $N$ is the number of symbols in one frame, and $\gamma_{S \rightarrow D(i)}$ and $\gamma_{R \rightarrow D(i)}$ are the instantaneous SNR’s when transmitting the $i$-th symbol as the local and relayed data, respectively. This is plotted along with the FERs in Figure 3.13.

3.7 Summary

One of the “themes” of [45] and [46] is that, when two partners are serving as relays for each other, it is possible to exploit the fact that each knows what the other is relaying and in doing so enhance the partner-to-partner link. That theme, which was first applied to uncoded bandwidth-efficient cooperation in [48], was applied in this chapter to coded modulation. Specifically, it was shown that the cooperative performance of LDPC-encoded QAM transmission can be substantially improved through the use of Ungerboeck-inspired “set partitioning” that improves the partner-to-partner link.
and thereby increases the likelihood that cooperative diversity is available at the destination. Moreover, that same set partitioning can be exploited at the destination by allowing the decoder to leverage past (or future) decoding success to improve the likelihood of current success. We then used outage analysis to demonstrate the efficacy of set-partitioning based cooperation over the more conventional approach in which the transmission of local data and relayed data are separated in time.
4.1 Introduction

In this chapter, we continue to consider the typical cooperative communication scenario as shown in Figure 3.1, where two source nodes/partners A and B assist each other to convey their data to one common destination Node D. Recall that the main difference between this system and a three-node relay system [28] is that, in the former, Nodes A and B act as both a source and a relay, i.e., they each transmit newly-generated local information and also re-transmit data containing the partner’s local information, whenever possible, whereas the latter contains a pure source, a pure relay, and a destination.

As previously described in Chapter 3, a variant of the decode-and-forward protocol in [39], called coded cooperation [44], incorporates channel coding into relaying by having each source transmit additional parity-check bits for its partner instead of simply repeating its partner’s data. In this way, the destination benefits from information accumulation rather than SNR accumulation and thus achieves a lower outage probability compared with a repetition-based system.

Practical rate-compatible codes are obviously good candidates for use in coded cooperation, since it is straightforward to partition a low-rate codeword into two parts, i.e., a high-rate codeword and additional parity-check bits, and adjust the resource allocation between source and relay. Rate-compatible convolutional codes
were used in [44] for a coded cooperation system, while [53] focused on the design of rate-compatible low-density parity-check (LDPC) block codes to achieve a diversity order of two. In this chapter, rate-compatible spatially coupled LDPC (SC-LDPC) codes are used for coded cooperation to exploit their capacity-achieving performance on general binary memoryless symmetric (BMS) channels.

We have briefly reviewed the literature of SC-LDPC codes in the introduction part of Chapter 2. In addition, Si et al. [54] proposed construction a family of rate-compatible regular SC-LDPC codes based on graph extension [55] and proved that all the code ensembles of different rates in the family provide capacity-achieving performance on the binary erasure channel (BEC). The same authors also proposed bilayer SC-LDPC codes for a three-node relay system in [23], which were able to provide the highest possible transmission rate for decode-and-forward relaying and simultaneously achieve capacities of both the source-to-relay channel and the source-to-destination channel. Again, for the three-node model, Uchikawa et al. [28] designed several protograph-based SC-LDPC code ensembles for the binary erasure relay channel, and spatially coupled versions of the MacKay-Neal codes were shown to achieve the capacity of this channel. For other types of relay systems, in [56], Schwandter et al. designed bilayer SC-LDPC codes for a network with two sources, one relay and one destination, also on the binary erasure relay channel, and these codes achieved capacity under the symmetric channel condition and approached capacity under the asymmetric channel condition.

Our main contribution in this chapter is to extend the use of SC-LDPC codes to coded cooperation systems in which:

1. When serving as a relay, the transmitter switches intelligently between the cooperative and non-cooperative transmission modes, and

2. Signals are corrupted by block fading with complex-valued additive white Gaussian noise (AWGN).
In what follows, Section 4.2 reviews the system model, and Section 4.3 briefly describes the structure of rate-compatible protograph-based spatially coupled LDPC code families. Section 4.4 gives simulation results for the outage performance, and some conclusions are drawn in Section 4.5.

4.2 System Model

As shown in Figure 3.1, two sources Nodes A and B cooperate to convey their data to a common destination Node D. In this chapter, unlike Chapter 3, the original and conventional scheduling in [44] is used. Assume that time is slotted and each slot is further divided into two phases: a local phase followed by a relay phase. Within each phase, Node A first transmits, followed by Node B. In this way, each time slot comprises four transmissions, two in the local phase and two in the relay phase. Compared to Chapter 3, in this chapter, local and relayed data are not “combined”.

Without loss of generality, consider Node A’s transmission in a given time slot. During the first half of the local phase, Node A generates a new block of $K$ information bits and encodes them into an $N$-bit codeword $C_A$, using a rate-compatible block code. $C_A$ is comprised of two parts: $C_A = [C_{A,1}, C_{A,2}]$ where $C_{A,1} \in \{0, 1\}^{N_1}$, $C_{A,2} \in \{0, 1\}^{N_2}$, and $N_1 + N_2 = N$. Thus, $C_{A,1}$ is a high-rate codeword embedded in the low-rate codeword $C_A$, and $C_{A,2}$ forms additional parity-check bits for $C_{A,1}$. Node A transmits the BPSK-modulated symbols of $C_{A,1}$, denoted as $x_{A,1} = \mod_{\text{BPSK}}(C_{A,1})$, to complete its role in the local phase; Node B then operates analogously.

During the relay phase, if Node A has successfully decoded the $K$ bits generated by Node B during the local phase, i.e., if $C_{B,1}$ can be successfully recovered by Node A, then Node A generates $C_{B,2}$ and transmits $x_{A,2} = \mod_{\text{BPSK}}(C_{B,2})$, operating in the cooperative transmission mode. Otherwise, Node A transmits its own additional parity check bits as $x_{A,2} = \mod_{\text{BPSK}}(C_{A,2})$, operating in the non-cooperative transmission mode. Figure 4.1 summarizes this “decoding-dependent” transmission
strategy and shows the transmitted codeword bits of Node A and B in one time slot.

Since BPSK is used for modulation, the overall spectral efficiency $\eta$ (bits/symbol) is equal to $k/n$, the rate of the low-rate code. As previously mentioned in Chapter 3, a key parameter for coded cooperation, called the “cooperation level”, is defined as

$$\beta = \frac{N_2}{N}.$$ 

Corollary 1 in [53] states a necessary condition for a two-user coded cooperation system to achieve full diversity:

$$\eta \leq \min\{\beta, 1 - \beta\}.$$ 

The channel model is the same as the one used in Chapter 3: all four channels – the two source-to-relay channels A-to-B and B-to-A, and the two source-to-destination channels A-to-D and and B-to-D – are corrupted by complex-valued AWGN and block Rayleigh fading:

- The average signal-to-noise ratio of all the channels are the same.

- For each channel, the block fading factor remains constant during an entire time
slot, and from one time slot to another, the fading factor varies independently
with the same probability distribution.

- Channels are independent of each other, and the distributions of the fading
factors are the same.

We still assume that perfect channel information (the realized fading factor and the
average noise variance $N_0$) is available at the receiver, but not at the transmitter.

Similar to those cooperative approaches in Chapter 3, there are four states in
which Node D’s receiver can operate, as shown in Figure 4.2, where state $S_0$, $S_1$, $S_2$,
and $S_3$ are listed for Node D’s decoding of Node A’s data as an example.

Let $y_{S,i}$, $S \in \{A, B\}$ and $i \in \{1, 2\}$, denote the block-faded, noisy version of $x_{S,i}$.

Similar to previously described in Chapter 3,

1. State $S_0$: Nodes A and B both transmitted in the non-cooperative mode. There
is no cooperative diversity achieved for Node A, and Node D decodes $C_A$ based
on $y_A = [y_{A,1}, y_{A,2}]$, i.e., the concatenated received symbols from Node A in
both the local and the relay phases. Moreover, decoding of Node B’s $C_B$ is also
in state $S_0$.

2. State $S_1$: Node A transmitted in the non-cooperative mode, but Node B trans-
mittted in the cooperative mode. Now there is cooperative diversity, and decoding
of $C_{A,2}$ is enhanced by maximal ratio combining (MRC) of $y_{A,2}$ and $y_{B,2}$.
Since both Nodes A and B transmitted for Node A in the relay phase, decoding
of Node B is in state $S_2$ as follows.

3. State $S_2$: Node A transmitted in the cooperative mode, but Node B transmitted
in the non-cooperative mode. Again, there is no cooperative diversity for Node
A’s data, but unlike state $S_0$, Node D decodes the high-rate codeword $C_{A,1}$
based only on $y_{A,1}$. However, in this case, cooperative diversity is achieved for
\[ \mathcal{S}_0 \]
\[
\begin{align*}
    x_{A,1} &= \text{mod}_{\text{BPSK}}(C_{A,1}) \\
    x_{B,1} &= \text{mod}_{\text{BPSK}}(C_{B,1}) \\
    x_{A,2} &= \text{mod}_{\text{BPSK}}(C_{A,2}) \\
    x_{B,2} &= \text{mod}_{\text{BPSK}}(C_{B,2})
\end{align*}
\]

\[
\mathcal{S}_1 \]
\[
\begin{align*}
    x_{A,1} &= \text{mod}_{\text{BPSK}}(C_{A,1}) \\
    x_{B,1} &= \text{mod}_{\text{BPSK}}(C_{B,1}) \\
    x_{A,2} &= \text{mod}_{\text{BPSK}}(C_{A,2}) \\
    x_{B,2} &= \text{mod}_{\text{BPSK}}(C_{B,2})
\end{align*}
\]

\[
\mathcal{S}_2 \]
\[
\begin{align*}
    x_{A,1} &= \text{mod}_{\text{BPSK}}(C_{A,1}) \\
    x_{B,1} &= \text{mod}_{\text{BPSK}}(C_{B,1}) \\
    x_{A,2} &= \text{mod}_{\text{BPSK}}(C_{A,2}) \\
    x_{B,2} &= \text{mod}_{\text{BPSK}}(C_{B,2})
\end{align*}
\]

\[
\mathcal{S}_3 \]
\[
\begin{align*}
    x_{A,1} &= \text{mod}_{\text{BPSK}}(C_{A,1}) \\
    x_{B,1} &= \text{mod}_{\text{BPSK}}(C_{B,1}) \\
    x_{A,2} &= \text{mod}_{\text{BPSK}}(C_{A,2}) \\
    x_{B,2} &= \text{mod}_{\text{BPSK}}(C_{B,2})
\end{align*}
\]

Figure 4.2: Four possible states \( \mathcal{S}_i, i = 0, 1, 2, \) and 3 of Node D decoding Node A's local data in coded cooperation, depending on transmission of Node A and B in the local phase and the relay phase.
Node B, whose data is decoded by Node D in state \( S_1 \). So, from the perspective of the whole system, states \( S_1 \) and \( S_2 \) always occur in pairs.

4. State \( S_3 \): Both Nodes A and B transmitted in the cooperative mode, and Node D decodes \( C_A \) based on \([y_{A,1,}, y_{B,2}]\). Again, there is cooperative diversity, since \( C_{A,1} \) and \( C_{A,2} \) experience independent block fading and complex-valued AWGN. And decoding of Node B is in state \( S_3 \) as well, so cooperative diversity is achieved for both sources.

To operate in the correct state, Node D must know the transmission modes of Nodes A and B during the relay phase. Again, in this chapter, we assume that Node D has access to the mode information it needs to determine the correct state.

4.3 Rate-Compatible SC-LDPC Code Families

In this section, we review the structure of three rate-compatible SC-LDPC code families; each code ensemble in each family is constructed based on protograph using the edge spreading technique. For concepts of protograph, block base (parity-check) matrix (“\( \mathbf{B} \)”), edge spreading, component base matrices (“\( \{ \mathbf{B}_i \}_{i=0}^w \)”), coupling width (“\( w \)”), coupling length (“\( L \)”), and spatially-coupled base matrix (“\( \mathbf{B}_{SC} \)”), refer to Chapter 2.

4.3.1 Rate-Compatible SC-LDPC Code Families

A rate-compatible SC-LDPC (RC-SC-LDPC) code family can be derived from a good rate-compatible LDPC block code (RC-LDPC-BC) family already reported in the literature, such as 1) the rate-compatible protograph-based Raptor-like (RC-PBRL) code family, 2) the rate-compatible punctured-node protograph-based Raptor-like (RC-PN-PBRL) code family, and 3) the (3, 9, 3, 6) rate-compatible root-LDPC (RC-R-LDPC) code family. Both 1) and 2) were reported in [57], and 3) was re-
ported in [53]; only 3) the RC-R-LDPC code family was designed specially for coded cooperation. Note that each block code family

1. Contains multiple protograph-based block code ensembles with different rates,

and

2. Uses the graph extension [55] technique to achieve rate compatibility.

That is, each block code family starts with a protograph-based highest-rate mother code ensemble, and generates lower-rate code ensembles by progressively adding extra parity variable nodes and check nodes to the protograph, where the extra parity variable nodes are only connected to the extra check nodes.

In this chapter, we develop the RC-PBRL, the RC-PN-PBRL, and the RC-R-LDPC code families into their spatially coupled counterparts as

- The rate-compatible spatially coupled protograph-based Raptor-like (RC-SC-PBRL),

- The rate-compatible spatially coupled punctured-node protograph-based Raptor-like (RC-SC-PN-PBRL), and

- The rate-compatible spatially coupled root-LDPC (RC-SC-R-LDPC) code families, respectively, by applying the edge spreading technique to each of the code ensembles in the block code family, i.e., every base component matrix in the base matrix of the highest-rate SC-LDPC code ensemble is extended to generate lower-rate SC-LDPC code ensembles. This procedure is illustrated in Figure 4.3, where the coupling width \( w = 1 \). Then, these spatially coupled code families are adopted in coded cooperation to improve the system performance.
Figure 4.3. Rate compatibility is achieved by graph extension in the RC-SC-LDPC code family; $w = 1$. Every base component matrix is extended, and the additional parity variable nodes are only connected to the additional check nodes.
4.3.2 The RC-SC-PBRL Code Family

A block code ensemble from the RC-PBRL code family [57] can be represented by the following block base matrix (using notation similar to [58]):

\[
B^{(k)} = \begin{bmatrix}
(B^{(1)})_{(c_1-b) \times c_1} & 0_{(c_1-b) \times (k-1)} \\
(B^{(e,2)})_{1 \times c_1} & I_{k-1} \\
& & \vdots \\
(B^{(e,k)})_{1 \times c_1} & & & \mathbf{I}_{c_k-b \times c_k}
\end{bmatrix},
\]

in which

- \(B^{(1)}\) of size \((c_1 - b)\) by \(c_1\) is the base matrix of the highest-rate mother code ensemble, with the design rate \(b/c_1\).

- \(k\) is the index for incremental redundancy (IR) steps, where \(k = 1\) is for the mother code ensemble, and \(I_{k-1}\) is the \((k-1) \times (k-1)\) identity matrix (\(I_0\) is simply empty). In the \(k\)-th IR step, \(k \geq 2\), a new lower-rate LDPC block code ensemble is generated by extending the block base matrix \(B^{(k-1)}\) from the previous step to the new \(B^{(k)}\). To be more specific, a new parity variable (the rightmost one in \(I_{k-1}\)) is added, which is connected only to certain variable nodes from the mother code ensemble via \((B^{(e,k)})_{1 \times c_1}\), i.e., this new parity variable is equal to the exclusive-or result of those \(B^{(1)}\) variables involved in the same new check.

\[
\begin{bmatrix}
(B^{(e,k)})_{1 \times c_1} & 0_{1 \times (k-2)} & 1
\end{bmatrix}.
\]

As pointed in [57], this structure is similar to the Raptor code; however, unlike the Raptor code, \((B^{(e,k)})_{1 \times c_1}\) is not random, but well designed and thus predetermined for each IR step, so this code ensemble is described as “Raptor-like”.

114
Obviously,

\[ c_k = c_1 + (k - 1), \]

and the design rate for the \( k \)-th IR step is

\[ R_k = \frac{b}{c_k} = \frac{b}{(c_1 + (k - 1))}. \]

As for the spatially coupled version, based on the RC-PBRL code family example introduced by [57], an RC-SC-PBRL code family with \( w = 1 \) was first presented in [58]. In this chapter, we follow the edge-spreading format therein, and spread \( B^{(k)} \) into \( B_0^{(k)} \) and \( B_1^{(k)} \) for all \( k \). In particular, the block base matrix of the mother code ensemble is given as

\[
B^{(1)} = \begin{bmatrix}
4 & 1 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 2 & 1 & 2 & 1 & 2 \\
\end{bmatrix}
\]

with the design rate \( R_1 = 6/8 \), which is spread into

\[
B_0^{(1)} = \begin{bmatrix}
3 & 0 & 1 & 0 & 1 & 1 & 0 & 2 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\end{bmatrix}
\]

and

\[
B_0^{(1)} = \begin{bmatrix}
1 & 1 & 0 & 2 & 0 & 1 & 1 & 0 \\
0 & 2 & 1 & 0 & 2 & 0 & 1 & 1 \\
\end{bmatrix}.
\]

Moreover, there are totally 12 IR steps (including the mother code ensemble, \( k = 1 \)) in the RC-PBRL code family, and the design rates are \( R_k = 6/9, 6/10, \ldots, 6/18 \) for \( k = 2, 3, \ldots, 11 \). The \( \{B^{(e,k)}\}_{k=2}^{11} \) set connecting the additional parity variables (i.e.,
those in $I_{k-1}$ to $B^{(1)}$ is given as

\[
\begin{bmatrix}
B^{(e,2)} \\
B^{(e,3)} \\
\vdots \\
B^{(e,11)}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix},
\]

which is spread into

\[
\begin{bmatrix}
B_0^{(e,2)} \\
B_0^{(e,3)} \\
\vdots \\
B_0^{(e,11)}
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}
\]
Finally, for edge spreading of $I_{k-1}$, the rightmost entry “1” for the $k$-th IR step, $k \geq 2$, is randomly assigned to either $B_{0}^{(k)}$ or $B_{1}^{(k)}$, as long as no degree-one check is introduced.

As a result, when the coupling length is $L$, the design rates of the code ensembles in the RC-SC-PBRL code family are

\[
R_{k}^{(SC)} = 1.0 - \frac{(c_{k} - b)(L + m_{s})}{c_{k}L}
\]

for $2 \leq k < 10$, and

\[
R_{k}^{(SC)} = 1.0 - \frac{(c_{k} - b)(L + m_{s}) - 1}{c_{k}L}
\]

for $k = 10$, and

\[
R_{k}^{(SC)} = 1.0 - \frac{(c_{k} - b)(L + m_{s}) - 2}{c_{k}L}
\]
when $k = 11$. The “−1” term for $k = 10$ and the “−2” term for $k = 11$ are due to the all-zero check(s) at the end of the spatially coupled base matrix, which results in a small rate loss in addition to the rate loss due to termination – both these two types of rate loss vanish as the coupling length goes to infinity, i.e., $R_k^{(SC)} \rightarrow R_k$ for all $k$ when $L \rightarrow \infty$.

4.3.3 The RC-SC-PN-PBRL Code Family

We also develop the RC-PN-PBRL (“punctured-node”) block code family in [57] into a spatially coupled version as an RC-SC-PN-PBRL code family with the coupling width $w = 1$, in a similar way as above. In particular, the block base matrix of the mother code ensemble

$$B^{(1)} = \begin{bmatrix} 2 & 1 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

is spread into

$$B_0^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 2 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$B_1^{(1)} = \begin{bmatrix} 1 & 1 & 0 & 2 & 0 & 1 & 1 & 0 \\ 0 & 2 & 1 & 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Moreover,
1. $\{B_i^{(e,k)}\}_k^{11}$ for $i = 0, 1$ are the same as those of the RC-SC-PBRL code family, in order to make the performance comparison as fair as possible, and

2. The high-degree first variable node in every $B_i^{(k)}$ matrix is always punctured, consistent with the puncture pattern of the original RC-PN-PBRL block code family.

4.3.4 The (3,9,3,6) RC-SC-R-LDPC Code Family

Duyck et al. [53] focused on the application of LDPC block codes in a two-user coded cooperation system, and proposed a rate-compatible root-LDPC (RC-R-LDPC) coding structure that incorporates root-LDPC coding into the rate-compatible structure based on graph extension, achieving full diversity.

An RC-R-LDPC code family contains one high-rate code ensemble for decoding at the relay/the other partner, and one low-rate code ensemble for decoding at the destination. For finite-length codes, the parity-check matrix of the high-rate code is embedded in the parity-check matrix of the low-rate code. In particular, the low-rate parity-check matrix can be represented as

$$H = \begin{bmatrix}
H_S & 0 & 0 & 0 \\
0 & 0 & 0 & H_R \\
I & 0 & 0 & H_{2i} \\
H_{1i} & H_{1p} & 0 & I \\
\end{bmatrix},$$

where "$H_S$" is the high-rate parity-check matrix. Codeword bits corresponding to the left half of $H$ are transmitted by the source ("S"), while bits corresponding to the right half are transmitted by the relay ("R"). $I$ denotes the identity matrix (or more generally, a permutation matrix), which connects the source half to the relay half, and vice versa. Subscript $i$ is for "information" bits, so all the information bits
are split into two parts; subscript \( p \) is for “parity” bits. Note that \( H_{1i} \) and \( H_{1p} \) are jointly designed; so are \( H_{2i} \) and \( H_{2p} \).

Similar to Section 4.2, take the transmitting and receiving of Node A’s data as an example; then the source is Node A itself, while the relay is Node B. Let \( C \) denote codeword bits, and let, for example, \( C^A(H_S) \) denote Node A’s codeword bits corresponding to the \( H_S \) part in \( H \) during a given time slot. As above, let \( i \) denote the information bits. Then the transmitting and receiving process is as follows:

- During Node A’s transmission in the local phase of the time slot:
  
  - \( C^A(H_{1p}) \) is generated based on all the information bits \( i^A \), which contains \( C^A(H_{1i}) \) and \( C^A(H_{2i}) \);
  
  - \([C^A(H_{1i}), C^A(H_{1p})]\) are encoded into \( C^A(H_S) \) using \( H_S \), corresponding to the “\( C_1^A \)” in Section 4.2, and
  
  - \( C^A(H_S) \) is broadcast by Node A to B and D.

- If \( C^A(H_{1i}) \) and \( C^A(H_{1p}) \) are successfully recovered by Node B based on the high-rate \( H_S \), then in the relay phase, Node B works in the cooperative transmission mode, in which
  
  - \( C^A(H_{2i}) \) is generated based on \( C^A(H_{1i}) \) and \( C^A(H_{1p}) \);
  
  - \( C^A(H_{2p}) \) is generated based on \( C^A(H_{1i}) \) and \( C^A(H_{2i}) \);
  
  - \([C^A(H_{2i}), C^A(H_{2p})]\) are encoded into \( C^A(H_R) \) using \( H_R \), corresponding to the “\( C_2^A \)” in Section 4.2, and
  
  - \( C^A(H_R) \) is transmitted to Node D by Node B.

- Node D has received \( C^A(H_S) \) from Node A and \( C^A(H_R) \) from Node B. Information bits \( i^A \), consisting of \( C^A(H_{1i}) \) and \( C^A(H_{2i}) \), are to be recovered based on the whole low-rate \( H \), and any information bit error results in a frame error...
of Node A. The system performance is evaluated as the frame error rate (FER) at Node D, averaging between A and B over all the time slots.

And if Node B works in the non-cooperative mode, codeword bits of its own corresponding to $H_R$ are transmitted in the relay phase, and no cooperative diversity is achieved for Node A at Node D.

One simple example of the RC-R-LDPC code family is the $(3, 9, 3, 6)$ RC-R-LDPC code family proposed in [53], whose low-rate block base matrix is

$$B = \begin{bmatrix}
3 & 3 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 3 & 3 \\
1 & 0 & 0 & 2 & 3 & 0 \\
2 & 3 & 0 & 1 & 0 & 0
\end{bmatrix}, \quad (4.1)$$

and the \[3 \ 3 \ 3\] in the upper-left corner is the high-rate base matrix. Again, for Node A’s transmission in one time slot, all the information bits $i^A$ (corresponding to the 1st and the 4th variables in $B$, “1V” and “4V”) are encoded into a rate 2/3 codeword (corresponding to the 4th check, “4C”). Then, half of $i^A$ (1V) along with all the parity-check bits (2V) are further encoded into $C_1^A$ using a rate 2/3 code (1C) and transmitted. At Node B, $C_1^A$ (1C) is decoded, and then the other half of $i^A$ (4V) are generated (via 4C). Similar to the operations at Node A, using two rate 2/3 codes (3C followed by 2C), Node B transmits this half to Node D via $C_2^A$. At Node D, a codeword $C^A = [C_1^A, C_2^A]$ of overall rate 1/3 is decoded. Again, any bit error in $i^A$ (1V and 4V) results in a frame error of Node A.

For more details of the $(3, 9, 3, 6)$ RC-R-LDPC code family, see Section IV in [53], especially Figures 8 and 9.
The above $B$ in 4.1 can be spread into

$$B_0 = \begin{bmatrix}
2 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 2 \\
1 & 0 & 0 & 1 & 2 & 0 \\
1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}$$

and

$$B_1 = \begin{bmatrix}
1 & 2 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
1 & 2 & 0 & 1 & 0 & 0
\end{bmatrix}$$

to construct an RC-SC-R-LDPC code family with $w = 1$, and into

$$B_0 = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix},$$

$$B_1 = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix},$$
and

\[
B_2 = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

to construct a code family with \( w = 2 \). Similar to the RC-R-LDPC code family, when an RC-SC-R-LDPC code family is used for two-user coded cooperation, the “target” bits during decoding are only the information bits.

4.4 Outage Performance

In this section, we evaluate the outage performance of the coded cooperation system when the rate-compatible spatially coupled LDPC code families are applied.

We define two types of outage events. For a given channel – either a point-to-point channel (i.e., the source-to-relay channel) or a multiple access channel (consisting of the source-to-destination and the relay-to-destination channels, from the destination perspective),

1. A \textit{code outage} event occurs when the belief propagation threshold \( \theta \) of the LDPC code ensemble (either block or spatially coupled) used on the channel falls below the instantaneous signal-to-noise ratio (SNR) of the channel. Here, \( \theta \) is calculated via density evolution using the reciprocal channel approximation (RCA) method; for a protograph-based code ensemble, protograph-based RCA is used, which, as described in [14], takes into account both the degree distribution and the edge connections of the protograph.

2. A \textit{capacity outage} event occurs when the instantaneous channel capacity for the BPSK modulation falls below the spectral efficiency \( \eta \).
The system outage ("O") probability at the destination Node D is given by

\[ P_O = \sum_{i=0}^{3} P_O(S_i) \cdot P(S_i), \]

where \( P_O(S_i) \) is the conditional outage probability for state \( S_i \) and \( P(S_i) \) is the probability of \( S_i \). Since \( S_i \) is determined by the transmission modes of Nodes A and B in the relay phase, \( P(S_i) \) clearly depends on the outage behavior when A and B decode each other’s data sent in the local phase.

Our focus is on the system outage probability for the rate-compatible spatially coupled LDPC code families; let the coupling length \( L = 100 \). For comparison, the system outage probability for the corresponding rate-compatible LDPC block code families and the system code outage probability are also evaluated.

4.4.1 Using the RC-SC-PBRL and the RC-SC-PN-PBRL Code Families

For the RC-SC-PBRL code family, the \( k = 11 \) code ensemble with design rate \( R_{k}^{(SC)} = 590/1800 = 0.3278 \) is selected as the low-rate code ensemble for decoding at the destination Node D, indicating that the overall spectral efficiency \( \eta = 0.3278 \). The \( k = 2, 3, 4, 5, \) and \( 6 \) code ensembles are used as the high-rate ensembles for transmission during the local phase and thus decoding at the the other partner, so the cooperation levels are \( \beta = 5/9, 1/2, 4/9, 7/18, \) and \( 1/3 \), respectively. The same \( k \) parameters are used when the RC-SC-PN-PBRL code family is applied to the coded cooperation system, resulting in the same rates for both the high-rate and the low-rate code ensembles as in the RC-SC-PBRL case.

4.4.1.1 State-wise Threshold at the Destination: RC-SC-PBRL

Table 4.1 shows the AWGN channel belief-propagation thresholds (measured in symbol energy-to-noise ratio \( E_S/N_0 \)) of the RC-SC-PBRL code family 1) at a source
TABLE 4.1

THRESHOLDS USING THE RC-SC-PBRL CODE FAMILY
(MEASURED IN AWGN SYMBOL ENERGY-TO-NOISE RATIO $E_s/N_0$)

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$5/9$</th>
<th>$1/2$</th>
<th>$4/9$</th>
<th>$7/18$</th>
<th>$1/3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{\text{source}}$</td>
<td>1.1483</td>
<td>0.8745</td>
<td>0.7162</td>
<td>0.6109</td>
<td>0.5327</td>
</tr>
<tr>
<td>$\theta_{S_0}$</td>
<td>0.3139</td>
<td>0.3139</td>
<td>0.3139</td>
<td>0.3139</td>
<td>0.3139</td>
</tr>
<tr>
<td>$\theta_{S_2}$</td>
<td>1.1483</td>
<td>0.8745</td>
<td>0.7162</td>
<td>0.6109</td>
<td>0.5327</td>
</tr>
</tbody>
</table>

node, 2) in state $S_0$ at Node D, and 3) in state $S_2$ at Node D, for different values of $\beta$. In each of these three cases, data for decoding was received from a single point-to-point channel (either the source-to-relay or the source-to-destination channel), and since we assumed a block-fading model, the channel condition (i.e., the average SNR and the fading factor) remains the same for the entire time slot. Thus, $\theta_{\text{source}}$, $\theta_{S_0}$, and $\theta_{S_2}$ are all scalars, depending only on the channel from the source: the threshold is simply the minimum $E_s/N_0$ required for successful decoding on that single channel.

As expected:

1. $\theta_{S_0}$ remains constant as $\beta$ varies, because the decoding is always based on the same low-rate code ensemble in this state.

2. $\theta_{\text{source}} = \theta_{S_2}$, because in both cases the same high-rate code ensemble is used for decoding – at the other partner in the former and at the destination in the latter.

3. $\theta_{\text{source}}$ (and thus $\theta_{S_2}$) improves as $\beta$ decreases, because more resources are dedicated to transmitting local data when $\beta$ is smaller, i.e., a higher-rate (and thus more powerful) code ensemble is used during the local phase.
Figure 4.4. Thresholds of the $k = 11$ code ensemble in the RC-SC-PBRL code family in (a) state $S_1$ and (b) state $S_3$ at the destination Node D, compared with capacity of the BPSK modulation. Cooperation level $\beta = 1/2$, and spectral efficiency $\eta = 0.3278$. 

126
Figures 4.4(a) and 4.4(b) show $\theta_{S_1}$ for state $S_1$ and $\theta_{S_3}$ for state $S_3$, respectively, when $\beta = 1/2$. As previously mentioned, cooperative diversity is achieved in $S_1$, so essentially, the threshold is for the multiple access channel consisting of the source ("S")-to-destination ("D") and the relay ("R")-to-destination ("D") channels, i.e., $\theta_{S_1}$ is characterized as $(\theta_{S_1}(S), \theta_{S_1}(R))$. Given $\theta_{S_1}(S)$, i.e., the S-to-D $E_s/N_0$, $\theta_{S_1}(R)$ is the minimum R-to-D $E_s/N_0$ required for successful decoding at the destination. And $\theta_{S_3}$ is characterized in a similar way. For comparison, capacity of the BPSK modulation is also shown in Figure 4.4. It is observed that

1. In both states, the threshold is, of course, lower bounded by the capacity, and the gap between them is small and approximately uniform. This is an important property, especially for state $S_3$, because as the average SNR increases, $S_3$ becomes the most frequent state in which Node D operates, and the above property guarantees that the application of the RC-SC-PBRL code family provides a system code-outage probability close to the system capacity-outage probability, when the average SNR is moderate to large.

2. If $\theta_{S_1}(R) = \theta_{S_3}(R) = 0$, then $\theta_{S_1}(S) = \theta_{S_3}(S)$, because in both cases, the S-to-D channel has no contribution to decoding at the destination, and thus the decoding completely relies on the coded data from the R-to-D channel, where additional parity variables were transmitted.

3. If $\theta_{S_1}(R) = 0$, then $\theta_{S_1}(S) = \theta_{S_0}$. In this case, the destination received no data from the relay, and thus the decoding completely relies on the low-rate coded data from the S-to-D channel, just as in the case of state $S_0$.

4. If $\theta_{S_3}(R) = 0$, then $\theta_{S_3}(S) = \theta_{S_2}$. This is similar to the observation above; the decoding completely relies on the high-rate coded data from the S-to-D channel, just as in the case of state $S_2$. 

127
5. Obviously, when $\theta_{S_1}(R) = \theta_{S_3}(R)$, $\theta_{S_1}(S) \leq \theta_{S_3}(S)$; as a result, the conditional outage probability of $S_1$ is smaller than that of $S_3$. If we simply compare $S_1$ and $S_3$, it is true that the former is preferred to the latter, since in $S_1$ the additional parity variables were sent on both the S-to-D and the R-to-D channels and thus can be enhanced via maximal ratio combining, while in $S_3$ those variables were from the R-to-D channel only. However, recall that decoding Node A’s data in $S_1$ indicates decoding Node B’s data in $S_2$, and vice versa; $S_2$ is clearly the least preferred state, in which no cooperative diversity is achieved and the source data is protected only by the high-rate ensemble from the rate-compatible code family. Thus, the overall system performance is still sabotaged. On the other hand, when decodings of both sources are in $S_3$, cooperative diversity is achieved for them simultaneously.

Figures 4.5(a) and 4.5(b) show $\theta_{S_1}$ and $\theta_{S_3}$, respectively, when cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18). First, for state $S_1$ in Figure 4.5(a), it is observed that

1. When $\theta_{S_1}(S) = 0$, $\theta_{S_1}(R)$ improves as $\beta$ increases. In this case, the S-to-D channel has no contribution to decoding at the destination, and thus the decoding completely relies on the coded data from the R-to-D channel. So if more resource was dedicated to transmission of the relay, i.e., if $\beta$ is larger, then the decoding is obviously more reliable.

2. When $\theta_{S_1}(R) = 0$, $\theta_{S_1}(S) = \theta_{S_0}$ for all the $\beta$ values. This can be explained in the same way as observation 3 above for Figure 4.4(a). Decoding at the destination completely relies on the $k = 11$ low-rate code ensemble from the S-to-D channel; it does not matter how much transmission resource was dedicated to the relay, i.e., how $\beta$ varies, because the R-to-D channel never has any contribution in this case.
Figure 4.5. Thresholds of the $k = 11$ code ensemble in the RC-SC-PBRL code family in (a) state $S_1$ and (b) state $S_3$ at the destination Node D, when spectral efficiency $\eta = 0.3278$. Cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18).
And for state $\mathcal{S}_3$ in Figure 4.5(b), it is observed that

1. When $\theta_{\mathcal{S}_3}(S) = 0$, $\theta_{\mathcal{S}_3}(R)$ improves as $\beta$ increases. The reason is the same as for observation 1 above of $\theta_{\mathcal{S}_1}$ in Figure 4.5(a).

2. When $\theta_{\mathcal{S}_3}(R) = 0$, $\theta_{\mathcal{S}_3}(S)$ improves as $\beta$ decreases. In this case, the R-to-D channel has no contribution to decoding at the destination, and thus the decoding completely relies on the high-rate code ensemble from the S-to-D channel. So if more resource was dedicated to transmission of the source, i.e., if $\beta$ is smaller, then the decoding is obviously more reliable.

3. The thresholds for all different $\beta$ values cross at the point $E_s/N_0(S) = E_s/N_0(R) = \theta_{\mathcal{S}_0}$. At this point, the S-to-D and the R-to-D channels have the same instantaneous SNR, so they are statistically the same. As a result, the low-rate code ensemble was essentially transmitted on a single point-to-point channel for decoding at the destination: the destination virtually operates in state $\mathcal{S}_0$.

Note that although we use the RC-SC-PBRL code family as an example, the threshold properties summarized from Table 4.1, Figure 4.4, and Figure 4.5 are general for the coded cooperation system when different rate-compatible code families are applied.

4.4.1.2 System Outage Probability: Code vs. Capacity

We study the system code outage probability calculated at the destination, when the RC-SC-PBRL and the RC-SC-PN-PBRL (“punctured node”) code families are applied to the coded cooperation system:

- In each case, the $k = 11$ code ensemble is chosen as the low-rate ensemble.

- To show the benefit of introducing the spatial-coupling structure, performances of the corresponding block code families – RC-PBRL and RC-PN-PBRL – are
The system code outage probability is lower-bounded by the system code outage probability with the same spectral efficiency $\eta$, so the latter is evaluated as well. The RC-PBRL and the RC-PN-PBRL code families have $\eta = 1/3 \approx 0.3333$, and the RC-SC-PBRL and the RC-SC-PN-PBRL code families have $\eta = 590/1800 \approx 0.3278$, where the rate loss is due to the finite coupling length $L = 100$.

Unlike Figure 4.4 and 4.5 where the threshold is represented in terms of $E_s/N_0$, it is more convenient to use $E_b/N_0$ in dB for the system outage probabilities as the average SNR scale.
Figure 4.6 shows the system code outage probabilities when the RC-PBRL, the RC-PN-PBRL, the RC-SC-PBRL, and the RC-SC-PN-PBRL code families are applied, respectively, along with the corresponding system capacity outage probabilities of the BPSK modulation; cooperation level $\beta = 1/2$. It is observed that

1. As expected, introducing the spatially coupled structure improves the system code outage probability. A spatially coupled code ensemble has the threshold saturation [17, 18, 19] property and generally provides a better threshold compared with the corresponding block code ensemble, so from the perspective of the system outage probability, i.e., asymptotically, using the rate-compatible spatially coupled code family consisting of spatially coupled code ensembles is a better choice for the coded cooperation system. Table 4.2 summarizes the average $E_b/N_0$ in dB required to achieve system outage probability $P_O = 10^{-2}$. We can see that RC-SC-PBRL outperforms RC-PBRL by (approximately) 0.60 dB, while RC-SC-PN-PBRL outperforms RC-PN-PBRL by 0.54 dB.

2. Table 4.2 also shows that RC-SC-PN-PBRL outperforms RC-SC-PBRL. This is consistent with the fact that RC-PN-PBRL provides a better threshold than RC-PBRL for the same coding rate [57]. However, by introducing a punctured node, RC-PN-PBRL outperforms RC-PBRL by 0.15 dB, while this type of improvement is only 0.10 dB comparing RC-SC-PN-PBRL and RC-SC-PBRL.

3. The gaps between the RC-SC-PBRL and RC-SC-PN-PBRL code outage probabilities and the BPSK capacity outage probability (for $\eta = 0.3278$) are approximately uniform, when the average $E_b/N_0$ is moderate to large.

4. When the coupling length $L \to \infty$, $\eta \to 0.3333$ for RC-SC-PBRL and RC-SC-PN-PBRL. Take RC-SC-PBRL as an example.

- Simulation results show that the state-wise thresholds at the destination (as shown in Table 4.1 and Figures 4.4, for $\beta = 1/2$) do not further
TABLE 4.2

FOR FIGURE 4.6: AVERAGE SNR $E_b/N_0$ (DB)

WHEN THE SYSTEM OUTAGE PROBABILITY $P_O = 10^{-2}$; $\beta = 1/2$

<table>
<thead>
<tr>
<th></th>
<th>$E_b/N_0$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC-PBRL</td>
<td>15.27</td>
</tr>
<tr>
<td>RC-SC-PBRL</td>
<td>14.67</td>
</tr>
<tr>
<td>RC-PN-PBRL</td>
<td>15.12</td>
</tr>
<tr>
<td>RC-SC-PN-PBRL</td>
<td>14.57</td>
</tr>
<tr>
<td>BPSK capacity, $\eta = 0.3333$</td>
<td>14.40</td>
</tr>
<tr>
<td>BPSK capacity, $\eta = 0.3278$</td>
<td>14.32</td>
</tr>
</tbody>
</table>

improve as $L$ increases from 100 to 200 to 500, i.e., the $L = 100$ thresholds measured in $E_s/N_0$ are numerically equal to the $L = \infty$ thresholds.

- As a result, we can use the $L = 100$ thresholds to calculate the $L = \infty$ system code outage probability. Since the SNR for outage probability is in terms of $E_b/N_0$ (dB), when $\eta$ increases from 0.3278 to 0.3333, the outage probability improves. However, simulation results show that this improvement is negligible, i.e., the $L = 100$ outage probability provides a very good estimate of that of $L = \infty$. Compared to the BPSK capacity outage probability for $\eta = 0.3333$, the gap is 0.27 dB.

We conclude that the rate-compatible spatially coupled code families offer near-capacity system outage probabilities with no loss in spectral efficiency, even though the corresponding block code families used for construction in the edge-spreading process were not designed specifically for the coded cooperation system.
Figure 4.7. Comparison of the system outage probability at Node D, when RC-SC-PBRL is applied: cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18). Spectral efficiency $\eta = 0.3278$.

Figure 4.7 shows the RC-SC-PBRL system code outage probabilities, when the cooperation level $\beta = 1/3$ (i.e., 6/18), 7/18, 4/9 (i.e., 8/18), 1/2 (i.e., 9/18), and 5/9 (i.e., 10/18); spectral efficiency $\eta = 0.3278$. For each $\beta$, the corresponding BPSK capacity outage probability is also illustrated for comparison. It is observed that

1. For RC-SC-PBRL, the outage probability improves as $\beta$ decreases.

2. For each $\beta$, the gap between the code and capacity outage probabilities is approximately uniform for moderate to large average $E_b/N_0$ (dB).

3. Based on Figure 4.7, Table 4.3 summarizes the average $E_b/N_0$ (dB) required for outage probability $10^{-2}$. We can see that, over the entire range of $\beta$, using the RC-SC-PBRL code family provides performance within 2.5% of the theoretical limit.
TABLE 4.3

FOR FIGURE 4.7: RC-SC-R-LDPC, AVERAGE SNR $E_b/N_0$ (DB)

WHEN THE SYSTEM OUTAGE PROBABILITY $P_o = 10^{-2}$; $\eta = 0.3278$

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>5/9</th>
<th>1/2</th>
<th>4/9</th>
<th>7/18</th>
<th>1/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC-SC-PBRL</td>
<td>15.60</td>
<td>14.67</td>
<td>13.94</td>
<td>13.69</td>
<td>13.67</td>
</tr>
<tr>
<td>BPSK Capacity</td>
<td>15.26</td>
<td>14.32</td>
<td>13.77</td>
<td>13.50</td>
<td>13.45</td>
</tr>
<tr>
<td>Gap</td>
<td>0.34</td>
<td>0.35</td>
<td>0.17</td>
<td>0.19</td>
<td>0.22</td>
</tr>
</tbody>
</table>

4.4.1.3 The RC-SC-PBRL System Code Outage Probability: When Windowed Decoding is Used

As previously described in Chapter 2, due to their (terminated) convolutional structure, SC-LDPC codes can be decoded using a windowed decoding (WD) scheme [34, 35], in which only those variable and check nodes within a decoding window of size $W$ are updated during each belief-propagation iteration, and the window shifts across the code graph until decisions have been made for all the variable nodes. Compared with treating the SC-LDPC code simply as a special kind of block code and carrying out the decoding process across the entire graph (i.e., the flooding-schedule decoding, FSD), WD reduces the decoding latency at a expense of a (negligible) loss in performance, depending on the choice of $W$. For details of WD, including the definition of $W$ and the scheduling, refer to Chapter 2.

Table 4.4 shows the average $E_b/N_0$ (dB) required for system code outage probability $10^{-2}$, when the RC-SC-PBRL code family is applied with WD; cooperation level $\beta = 1/2$, and spectral efficiency $\eta = 0.3278$. It is observed that RC-SC-PBRL with window size $W = 5$ already outperforms RC-PBRL with FSD. Also, for RC-SC-PBRL, the WD outage probability improves as $W$ increases, and $W = 10$ provides
TABLE 4.4

AVERAGE $E_b/N_0$ (DB) WHEN THE SYSTEM OUTAGE PROBABILITY $P_o = 10^{-2}$:

RC-SC-PBRL WITH WINDOWED DECODING, $\beta = 1/2$, AND $\eta = 0.3278$

<table>
<thead>
<tr>
<th></th>
<th>$E_b/N_0$</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK capacity, $\eta = 0.3278$</td>
<td>14.32</td>
<td>-</td>
</tr>
<tr>
<td>RC-SC-PBRL, WD, $W = 5$</td>
<td>15.15</td>
<td>0.83</td>
</tr>
<tr>
<td>RC-SC-PBRL, WD, $W = 10$</td>
<td>14.68</td>
<td>0.36</td>
</tr>
<tr>
<td>RC-SC-PBRL, FSD</td>
<td>14.67</td>
<td>0.35</td>
</tr>
<tr>
<td>RC-PBRL, FSD, $\eta = 0.3333$</td>
<td>15.27</td>
<td>-</td>
</tr>
</tbody>
</table>

performance nearly the same as the FSD, which, if treated as a special case of WD, has $W = L + m_s = 101$, i.e., all the variable and check nodes are covered by the decoding window simultaneously.

4.4.2 Using the RC-SC-R-LDPC Code Family

Figure 4.8 shows the system code outage probabilities with cooperation level $\beta = 1/2$, when two types of rate-compatible spatially coupled root-LDPC (RC-SC-R-LDPC) code families are applied: 1) $m_s = 1$, $L = 100$, and 2) $m_s = 2$, $L = 200$. Based on these settings, the spectral efficiency is always $\eta = 0.3267$. Both code families are developed from the RC-R-LDPC block code family with $\eta = 0.3333$, and the edge spreading formats can be found in previous Section 4.3.4. The capacity outage probabilities with $\eta = 0.3267$ and 0.3333 are also illustrated as lower bounds of performance of the RC-SC-R-LDPC code families – the former for finite coupling
Figure 4.8. Comparison of the system outage probability at Node D: RC-R-LDPC, RC-SC-R-LDPC with $m_s = 1$ and $L = 100$, and RC-SC-R-LDPC with $m_s = 2$ and $L = 200$. Cooperation level $\beta = 1/2$.

length, and the latter for infinite coupling length with no loss in spectral efficiency. And based on Figure 4.8, Table 4.5 shows the average $E_b/N_0$ (dB) values required for outage probability $10^{-2}$. Since $m_s = 2$ provides very limited improvement (approximately 0.02 dB) compared with $m_s = 1$, we use the RC-SC-R-LDPC code family with $m_s = 1$ as the example for our discussion. Thus, it can be observed that

1. Again, introducing the spatial coupling structure improves the system outage probability, compared with using the corresponding block code family. RC-SC-R-LDPC with $m_s = 1$ outperforms RC-R-LDPC by 0.52 dB.

2. The gap between the system code and capacity outage probabilities is approximately uniform for moderate to large average $E_b/N_0$ values.

3. When the coupling length $L \to \infty$, i.e., when there is no loss in spectral ef-
TABLE 4.5

FOR FIGURE 4.8: AVERAGE $E_b/N_0$ (DB)

WHEN THE SYSTEM OUTAGE PROBABILITY $P_O = 10^{-2}; \beta = 1/2$

<table>
<thead>
<tr>
<th></th>
<th>$\eta$</th>
<th>$E_b/N_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC-R-LDPC</td>
<td>0.3333</td>
<td>15.35</td>
</tr>
<tr>
<td>RC-SC-R-LDPC, $m_s = 1$, $L = 100$</td>
<td>0.3267</td>
<td>14.83</td>
</tr>
<tr>
<td>RC-SC-R-LDPC, $m_s = 2$, $L = 200$</td>
<td>0.3267</td>
<td>14.81</td>
</tr>
<tr>
<td>BPSK capacity</td>
<td>0.3333</td>
<td>14.31</td>
</tr>
<tr>
<td>BPSK capacity</td>
<td>0.3267</td>
<td>14.40</td>
</tr>
</tbody>
</table>

efficiency, RC-SC-R-LDPC provides outage probability within 3% of capacity ($\eta = 0.3333$).

4.5 Summary

In this chapter, we have investigated the use of rate-compatible spatially coupled LDPC code families in a two-user coded cooperation system. In particular, three types of protograph-based spatially coupled code families, developed from good block code families already reported in the literature using the edge spreading technique, were employed as examples. In each case, the results demonstrate that introducing the spatial coupling structure provides near-capacity system outage probability (calculated at the destination node), significantly better than using the corresponding block code family – both with a finite coupling length $L$ and as $L \to \infty$ when there is no loss in spectral efficiency. This kind of improvement is observed 1) whether the block code family was originally designed for coded cooperation or not, and 2)
even when the convolutional structure of the spatially coupled code ensembles is explored via windowed decoding, i.e., when belief propagation is localized by a decoding window and the decoding latency is reduced with negligible loss in performance.
CHAPTER 5

CONCLUSIONS

Our contributions regarding the three topics in wireless communications covered in this dissertation –

- Design of spatially coupled low-density parity-check (SC-LDPC) codes over GF(q) for windowed decoding,
- Bandwidth-efficient cooperative communication, and
- Rate-compatible SC-LDPC codes in binary coded cooperation

– have already been summarized at the end of Chapters 2, 3, and 4, respectively.

Not quite surprisingly, throughout the development of all the work above, the author has been questioned about the four core concepts in this dissertation by people from both the academic area and the wireless industry: protograph, non-binary (“q-ary” for q > 2), spatial coupling, and cooperation (or to be more general, relaying):

5.1 Protograph: Why Not Just Randomized Construction?

The key idea of using protographs is to make things “structured”: code design, encoding, and decoding. Protograph-based threshold analyzing tools can be used to search for protograph-based LDPC code ensembles with better thresholds than randomized/unstructured code ensembles with the same degree distribution [15]. As a special kind of protograph-based LDPC codes, quasi-cyclic codes can be encoded with feedback shift-registers, lead to more efficient decoder structure, and provide
impressive performance when the blocklength is moderate [12]. That is, when LDPC
codes are considered for application, protograph provides benefits all the way from
code design to practical implementation; in fact, nearly all the LDPC codes in cur-
rent wired and wireless communication standards can be viewed as protograph-based
LDPC codes.

5.2 Non-binary: Is It Worth the Complexity?

Yes. Although in Chapter 2 we study the belief-propagation decoding thresholds
of $q$-ary block and spatially-coupled LDPC code ensembles for $q = 2$ (binary) up
to $q = 1024$, we are not simply pursuing good performance by using a large-size
finite field without considering the issue of decoding complexity. As discussed in
Section 2.5 and in reference [37], 4-ary LDPC codes often outperform their binary
counterparts significantly with a small increase in complexity. Moreover, when com-
bined with bandwidth-efficient modulation schemes, non-binary LDPC codes show
huge performance gain, competing against bit-interleaved coded modulation and mul-
tilevel coding with binary LDPC codes; this has been well studied in the literature of
optical fiber communication, where both a steep waterfall region closed to threshold
and a very low error floor are required.

5.3 Spatial Coupling: Block Codes Are Not Good Enough?

We have no intent to start a “war” against LDPC block codes; many industry stan-
dards have chosen good LDPC block codes, such as IEEE 802.11n (wireless LANs)
and IEEE 802.16e (WiMax). To the author, spatial coupling is a philosophy; this
principle has been applied to areas other than channel coding, including compressed
sensing, statistical physics, etc. – refer to [29] and the references therein. When
facing a code design problem, one can reach out for SC-LDPC codes to circumvent
some special considerations needed for LDPC block codes. For example, Chapter 4
applied rate-compatible SC-LDPC codes directly to coded cooperation without designing the code ensembles specially for the cooperation system, and the theoretical performance turns out to be satisfactory. This kind of idea has also been reflected in code design for, for example, rate compatibility via puncturing [59] and block fading channels [60]. Moreover, the decoding latency problem for SC-LDPC codes have been solved by design of protograph-based code ensembles suitable for windowed decoding in this dissertation, thus making SC-LDPC codes even more promising. To the author’s knowledge, the use of SC-LDPC codes is spreading in the industry of, for example, flash memory.

5.4 Relaying: Is It Realistic?

Yes. In 3GPP, relay study started in January 2009, first kicked off by the physical layer group; LTE Release 10 (part of LTE-Advanced) proposed a decode-and-forward relay, although essentially, it is still a stand-alone eNodeB rather than a cooperative relay as shown in this dissertation. On the IEEE side, 802.16j supports a wide range of relay categories. Relaying has also been adopted, or at least considered, in cognitive radio networks, small cells, and 5G networks. Practical concerns, such as extra power consumption and more complicated scheduling, are being solved in the literature.

We hope that this dissertation may encourage the spreading of spatial coupling in code design for modern wireless communication systems.
BIBLIOGRAPHY


143


Information Theory, pp. 1474–1478, St. Petersburg, Russia, July 2011.


37. K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, Jr., “Performance comparison of non-binary LDPC block and spatially coupled codes,”


47. L. Jing, G. Jian Hua, Y. Wang, and H. Yao, “Performance analysis and partner selection for cooperative diversity based on MQAM modulation,” *Science China*,


