Novel LDPC coding and decoding strategies: design, analysis, and algorithms

This thesis is submitted in partial fulfilment of the requirements for

Doctor of Philosophy (Ph.D.)

Jingjing Liu

Communications Research Group

Department of Electronics

University of York

December 2012

Abstract

In this digital era, modern communication systems play an essential part in nearly every aspect of life, with examples ranging from mobile networks and satellite communications to Internet and data transfer. Unfortunately, all communication systems in a practical setting are noisy, which indicates that we can either improve the physical characteristics of the channel or find a possible systematical solution, i.e. error control coding. The history of error control coding dates back to 1948 when Claude Shannon published his celebrated work "*A Mathematical Theory of Communication*", which built a framework for channel coding, source coding and information theory. For the first time, we saw evidence for the existence of channel codes, which enable reliable communication as long as the information rate of the code does not surpass the so-called channel capacity. Nevertheless, in the following 60 years none of the codes have been proven closely to approach the theoretical bound until the arrival of turbo codes and the renaissance of LDPC codes. As a strong contender of turbo codes, the advantages of LDPC codes include parallel implementation of decoding algorithms and, more crucially, graphical construction of codes. However, there are also some drawbacks to LDPC codes, e.g. significant performance degradation due to the presence of short cycles or very high decoding latency. In this thesis, we will focus on the practical realisation of finite-length LDPC codes and devise algorithms to tackle those issues.

Firstly, rate-compatible (RC) LDPC codes with short/moderate block lengths are investigated on the basis of optimising the graphical structure of the tanner graph (TG), in order to achieve a variety of code rates $(0.1 < R < 0.9)$ by only using a single encoder-decoder pair. As is widely recognised in the literature, the presence of short cycles considerably reduces the overall performance of LDPC codes which significantly limits their application in communication systems. To reduce the impact of short cycles effectively

for different code rates, algorithms for counting short cycles and a graph-related metric called Extrinsic Message Degree (EMD) are applied with the development of the proposed puncturing and extension techniques. A complete set of simulations are carried out to demonstrate that the proposed RC designs can largely minimise the performance loss caused by puncturing or extension.

Secondly, at the decoding end, we study novel decoding strategies which compensate for the negative effect of short cycles by reweighting part of the extrinsic messages exchanged between the nodes of a TG. The proposed reweighted belief propagation (BP) algorithms aim to implement efficient decoding, i.e. accurate signal reconstruction and low decoding latency, for LDPC codes via various design methods. A variable factor appearance probability belief propagation (VFAP-BP) algorithm is proposed along with an improved version called a locally-optimized reweighted (LOW)-BP algorithm, both of which can be employed to enhance decoding performance significantly for regular and irregular LDPC codes. More importantly, the optimisation of reweighting parameters only takes place in an offline stage so that no additional computational complexity is required during the real-time decoding process.

Lastly, two iterative detection and decoding (IDD) receivers are presented for multiple-input multiple-output (MIMO) systems operating in a spatial multiplexing configuration. QR decomposition (QRD)-type IDD receivers utilise the proposed multiple-feedback (MF)-QRD or variable-M (VM)-QRD detection algorithm with a standard BP decoding algorithm, while knowledge-aided (KA)-type receivers are equipped with a simple soft parallel interference cancellation (PIC) detector and the proposed reweighted BP decoders. In the uncoded scenario, the proposed MF-QRD and VM-QRD algorithms are shown to approach optimal performance, yet require a reduced computational complexity. In the LDPC-coded scenario, simulation results have illustrated that the proposed QRD-type IDD receivers can offer near-optimal performance after a small number of detection/decoding iterations and the proposed KA-type IDD receivers significantly outperform receivers using alternative decoding algorithms, while requiring similar decoding complexity.

Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Rodrigo C. de Lamare, for his enthusiastic support, excellent guidance and continuous encouragement throughout my Ph.D. study at the University of York. I also wish to thank Dr. Henk Wymeersch from Chalmers University of Technology for providing precious technical advice during our collaboration.

I am grateful to all colleagues and staffs in Communications Research Group for their help and support. Special thanks are due to Dr. Peng Li and Li Li (Alex) for their invaluable friendship and many great technical discussions that we have had.

Last but not least, I would like to thank my family and friends who have been with my at each step, bringing so much joy and happiness into my life.

This thesis is dedicated to my parents. Their endless love and unconditional support have always been my source strength to go through difficulties and my motivation to achieve success.

Declaration

Elements of the research presented in this thesis have resulted in some publications. A list of thoese publications can be found at the end of Chapter 1.

All work presented in this thesis as original is so, to the best knowledge of the author. References and acknowledgements to other researchers have been given as appropriate.

Chapter 1

Introduction

Contents

1.1 Overview

It has been over 70 years since Claude Shannon published his famous "A Mathematical Theory of Communication", the foundation of the fields of channel coding, source coding and information theory, in which Shannon proved the existence of channel codes that are able to provide reliable communication as long as the code rate does not exceed the so-called channel capacity [1]. Following his seminal work and the framework he developed, over the next few decades a great number of innovative and efficient coding schemes were invented and applied in real-life communications systems. Despite the highly successful developed channel coding schemes, none of these codes has been able to approach the theoretical limit closely in practice [2]. During the 1990s, the situation changed dramatically with the invention of Turbo Codes [3] and the rediscovery of low-density parity-check

(LDPC) codes [4], both of which have been shown to have near-capacity performance. Since then, a rich state of the art with respect to the "turbo concept" and sparse graphical models have formed the basic framework of "modern coding theory", which still relies on Shannon's formulation but mainly concentrates on the random sparseness of code description and the nature of message passing algorithms [6].

Digital wireless communications are omnipresent in our daily lives, with examples ranging from mobile phones and digital television via satellite or terrestrial links to wireless Internet connections. In all those systems, coding schemes play an essential role in ensuring successful transmission of information, which is represented by a sequence of bits, from one point to another, [7] and [8]. In order to combat channel noise, a coding strategy is devised that can construct codewords by adding redundancy to the transmitted bits, such that the original information can be perfectly decoded even with a certain number of errors [9]. One of the most advanced classes of channel codes is the class of LPDC codes, which were first proposed by Gallager [10] in the early 1960s and rediscovered and generalized by MacKay et al. in the 1990s [4]. As strong competitors to Turbo Codes, LDPC codes are well known not only for their capacity-approaching performance but also for their manageable decoding complexity [11]. More importantly, LDPC codes have some of the advantages of linear block codes, such as their simplicity and sparse (low-density) parity-check matrices which can be depicted as a graphical model called a Tanner graph (TG) [12]. Graphical approaches are often preferred because they provide a means of visualising and analysing complex mathematical relationships [13].

Recently, the realization of various code rates and the achievement of good decoding performance have become central issues for error control coding schemes [6]. In this thesis we investigate these two problems with respect to finite-length LDPC codes and propose novel design concepts and accompanying encoding and decoding algorithms that build upon existing techniques in the fields of modern coding design, statistical inference, graph theory, and iterative receiver design. Firstly, rate-compatible (RC) LDPC codes, based on the optimization of the structure of TG, are studied to fulfill the need to implement different rates with a single pair of encoder/decoder when the data transmission occurs via time-varying channels. Since the presence of short cycles in TG is responsible for a performance degradation of LDPC codes [2], especially the error floor phenomenon [6], algorithms for counting cycles [15, 16] and a related metric called Ex-

trinsic Message Degree (EMD) [14], are considered in order to reduce the impact of short cycles for different code rates. An analysis and a comprehensive study are presented that illustrates the effectiveness of the proposed schemes in terms of improving the cycle distribution of TG.

Secondly, we have devised an LDPC decoding algorithm that can offer reliable transmission with a lower decoding delay, i.e. faster convergence. Iterative decoding algorithms that decode LDPC codes, are parallelizable in hardware and have much lower per-iteration complexity than developed Turbo Codes [9]. Nevertheless, these decoding algorithms are no longer efficient when decoding poorly-designed codes, i.e. codes which have too many short cycles (in particular cycles of length 4), for which a large number of iterations is required and the decoder may fail to recover the message. Thus, novel belief propagation (BP) algorithms are investigated which borrow the idea of reweighting the extrinsic message [17] in the process of message passing decoding. The proposed decoders are tested to decode practical LDPC codes using both random and combinatorial designs.

Lastly, an iterative detection and decoding (IDD) strategy is presented, which combines low complexity detectors with the proposed LDPC decoding algorithm for multiple-input multiple-output (MIMO) systems operating with a spatial multiplexing configuration. Shortly after the advent of turbo codes, an advanced IDD framework was reported in [18] and [23] with affordable computational complexity. Since then, the "turbo concept" has been verified to effectively solve detection and decoding problems in wireless communications. However, the selection of low complexity detectors and powerful channel codes is still an open question. Consequently, LDPC-coded IDD receivers were developed which can achieve performance comparable to that of the maximum-likelihood (ML) solution.

1.2 Contributions

The contributions of this thesis are summarized as follows:

- Novel design strategies for RC-LDPC codes suited to high code rates have been developed. In particular, we present two effective puncturing schemes which can mitigate the performance loss when compared to random puncturing or other intentional puncturing schemes. For the purpose of benchmarking, another puncturing scheme is also proposed which is on the basis of brute-force search. The puncturing patterns of the proposed schemes can be obtained off-line, i.e. independent of transmitted symbols. The first puncturing scheme employs cycle counting algorithms and grouping of the variable nodes having equal lengths of short cycles on Tanner Graph. The second scheme relies on the EMD metric, and the third scheme is a simulation-based exhaustive search to find the best puncturing pattern among several random puncturing patterns. We show that all the proposed approaches manage to generate RC-LDPC codes across a wide range of rates and meanwhile have good puncturing performance as compared to existing puncturing schemes. From the puncturing point of view, our simulation results show that the structure of short cycles is of fundamental importance to the overall performance of LDPC codes.

- To construct RC codes at low rates, two innovative extension schemes have been devised based on the structure of short cycles and the approximate cycle EMD (ACE) spectrum [19]. The proposed techniques are able to create RC-LDPC codes that are highly flexible in block length, regularity (available for both regular and irregular codes) and code rates. A comprehensive study of RC-LDPC codes is carried out on the basis of cycle distribution and the ACE spectrum. A set of puncturing/extension strategies is considered to create RC-LDPC codes with a wide range of rates from 0.1 to 0.9. Simulation studies including an extensive set of comparisons with previously reported algorithms are conducted. The performance of RC-LDPC codes is shown in terms of bit-error rate (BER), frame-error rate (FER) and throughput for a type-II hybrid automatic repeat-request (ARQ) system.

- We investigate the idea of reweighting a suitable part of the factorized graph while also taking the effect of short cycles into account. By combining a reweighting strategy with knowledge of short cycles, a novel BP algorithm, variable factor appearance probabilities or (VFAP)-BP, is presented that assigns distinct reweighting values to each parity-check node on the basis of the structure of short cycles rather than complex global graphical optimization. We also extend the application

of reweighted message passing decoding algorithms from symmetric to asymmetric graphs. Simulation results show that the proposed decoding algorithm consistently outperforms uniformly reweighted (URW)-BP [20] for irregular LDPC codes, and offers better BER performance than standard BP for both regular and irregular codes when using a small number of iterations.

- After further study on graphical models and variational inference, a more powerful decoding algorithm is presented which is called the locally-optimized reweighted or (LOW)-BP algorithm. Based on the framework of [17], we successfully transform intractable global optimization on FAPs, due to the size of the graph, into a tractable local optimization problem by separating the TG into a number of subgraphs in which the resulting FAPs are locally optimal. The course of graph expansion is realized by slightly adjusting the progressive edge-growth (PEG) technique reported in [21]. In fact, our optimization can be interpreted as a variational problem that maximizes the mutual information in each subgraph while optimizing FAPs with the conditional gradient method. Numerical results show that LOW-BP considerably improves the convergence behavior of the BP decoding algorithm, which allows lower decoding latency.

- We propose IDD schemes for MIMO systems equipped with novel knowledge-aided (KA)-BP algorithms, VFAP-BP and LOW-BP, that can considerably improve spatial multiplexing and diversity gains. Compared to the maximum *a posteriori* (MAP) solution, the complexity of the proposed LDPC-coded systems is largely reduced thanks to the application of simple detectors, such as linear minimum mean square error (MMSE), successive interference cancelation (SIC) and parallel interference cancelation (PIC). The analysis of EXIT charts demonstrates that the proposed decoders are well matched to low complexity detectors. Simulation results also show that the proposed decoding algorithms consistently outperform existing algorithms when performing IDD in multi-antenna systems with a small number of decoding iterations.

1.3 Thesis Outline

The rest of the thesis is organized as follows:

- Chapter 2 provides an overview of LDPC codes by presenting the fundamental concepts and theory behind these codes, including the representation of LDPC codes, construction methods, encoding procedure, and widely used decoding algorithms with distinct operations. Basic problems in channel coding and graph theory are also also considered in this context.

- Chapter 3 details three proposed puncturing schemes and two proposed extension schemes. A detailed system model is introduced along with a description of the proposed algorithms and a technical analysis. A performance comparison of RC-LDPC codes, including existing methods, is presented in terms of BER and FER as well as throughput.

- Chapter 4 presents the proposed VFAP-BP decoding algorithm. The motivation, technical background, and derivation of message-passing rules are described in detail. The decoding performance of the proposed VFAP-BP algorithm, along with a convergence analysis, is illustrated and thoroughly discussed.

- Chapter 5 presents the proposed LOW-BP decoding algorithm. The concept of locally optimizing FAPs is explained in detail, and a technical analysis is carried out to explore the fundamental differences between LOW-BP, VFAP-BP and URW-BP. Moreover, we consider a performance evaluation of LOW-BP against existing algorithms for both regular and irregular LDPC codes.

- Chapter 6 presents an IDD scheme for MIMO systems that incorporates distinct detectors with either VFAP-BP or LOW-BP decoding strategies. From the overall performance and EXIT charts analysis, it is obvious that the proposed LDPC-coded technique is capable of approaching near-ML performance by effectively mitigating inter-antenna interference and additive noise.

- In Chapter 7, conclusions are drawn and a discussion of the possibility of future work is presented.

1.4 Notation

In this thesis, we use capital and small bold fonts to denote matrices and vectors, such as $\boldsymbol{A}$ and $\boldsymbol{a}$. Elements of matrices and vectors are denoted as $a_{m,n} = [\boldsymbol{A}]_{m,n}$ and $a_m = [\boldsymbol{a}]_m$, respectively. Unless otherwise stated, the symbol j is an imaginary unit for which $j^2 = -1$. $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary components of a complex number, $(\cdot)^*$ denotes the complex conjugate, and $\boldsymbol{I}_Q$ denotes a $Q \times Q$ identity matrix. Moreover, $(\cdot)^T$ and $(\cdot)^H$ denote the matrix transpose and the Hermitian transpose, respectively. $\mathbb{E}\{\cdot\}$ denotes the statistical expectation operator and $\mathrm{tr}\{\cdot\}$ denotes the trace operator.

1.5 Publication List

Journal Papers

1. J. Liu, R. C. de Lamare, "Low-Latency Reweighted Belief Propagation Decoding for LDPC Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1660-1663, October 2012.

2. J. Liu and R. C. de Lamare, "Finite-Length Rate-Compatible LDPC Codes Based on Puncturing and Extension Techniques," *IET Communications*, 2013 (under review).

3. P. Li, R. C. de Lamare and J. Liu, "Adaptive Decision Feedback Detection with Parallel Interference Cancellation and Constellation Constraints for Multiuser MIMO systems," *IET Communications*, 2013 (accepted and awaiting publication).

4. J. Liu, R. C. de Lamare and H. Wymeersch, "Improved Knowledge-Aided Reweighted Belief Propagation Decoding Algorithms for Finite-Length LDPC codes," *IEEE Transactions on Communications*, 2013 (in preparation).

5. J. Liu, P. Li and R. C. de Lamare, "Novel Knowledge-Aided Reweighted Belief Propagation Algorithms for Iterative Detection and Decoding Multi-Antenna Systems," *IEEE Transactions on Vehicular Technology*, 2013 (in preparation).

Conference Papers

1. J. Liu, R. C. de Lamare and H. Wymeersch, "Locally-Optimized Reweighted Belief Propagation Algorithm for Decoding Finite-Length Irregular LDPC codes," *IEEE Wireless Communications and Networking Conference (WCNC 2013)*, Shanghai, China, April 2013 (accepted and awaiting publication).

2. J. Liu, P. Li and R. C. de Lamare, "Iterative Detection and Decoding for MIMO Systems with Knowledge-Aided Belief Propagation Algorithms," *46th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, November 2012 (accepted and awaiting publication).

3. J. Liu, R. C. de Lamare, "Knowledge-Aided Reweighted Belief Propagation Decoding for Regular and Irregular LDPC Codes with Short Blocks," *9th IEEE International Symposium on Wireless Communications Systems (ISWCS)*, pp. 984-988, Paris, France, August 2012.

4. P. Li, J. Liu and R. C. de Lamare, "Adaptive Iterative Decision Multi-Feedback Detection for Multiuser MIMO Systems," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pp. 3037-3040, Kyoto, Japan, March 2012.

5. J. Liu, R. C. de Lamare, "Finite-length rate-compatible LDPC codes based on extension techniques," *8th IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pp. 41-45, Aachen, Germany, November 2011.

6. J. Liu, P. Li, L. Li, R. C. de Lamare and A. G. Burr, "Iterative QR Decomposition-Based Detection Algorithms with Multiple Feedback and Dynamic Tree Search for LDPC-Coded MIMO Systems," *Sensor Signal Processing for Defence (SSPD 2011)*, pp. 1-5, London, UK, September 2011.

7. J. Liu, R. C. de Lamare, "Novel intentional puncturing schemes for finite-length irregular LDPC codes," *17th IEEE International Conference on Digital Signal Processing (DSP 2011)*, pp. 1-6, Corfu, Greece, July 2011.

Chapter 2

Literature Review

Contents

2.1 Introduction

This chapter introduces the background to LDPC codes and provides a foundation for further study of LDPC codes in later chapters. We start with an overview of the communication model formalized by Shannon and his famous channel capacity formula. Following that, representation methods of LDPC codes are presented alongside two classifications. We then summarize the design approaches to LDPC codes among which some well known construction algorithms are detailed. Effective tools used for asymptotic analysis are also included. At the end of the chapter, different versions of the BP algorithm are described for decoding LDPC codes with variable complexity and performance.

2.2 Channel Coding Overview

This thesis focuses on Shannon's classical problem [1]: how efficiently to transmit a message across a noisy channel such that the receiver can determine the message with high accuracy in spite of an imperfect channel. We aim to devise reliable channel coding schemes that offer capacity-approaching performance while introducing low delay or latency at the same time.

2.2.1 Classical Communication System Model

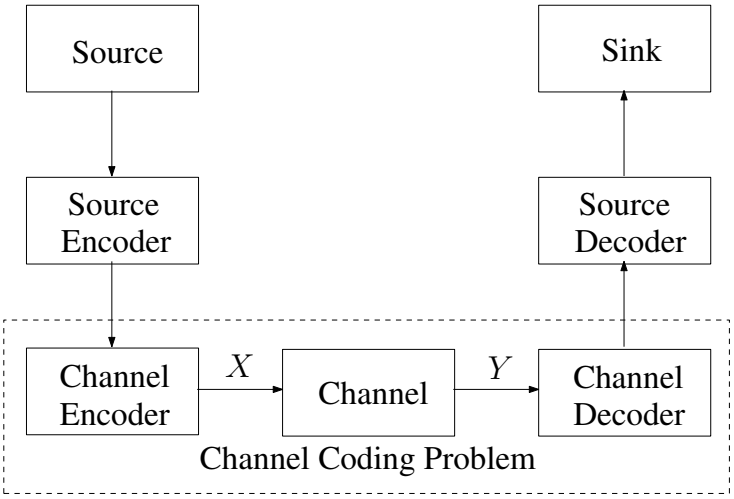

Figure 2.1: Shannon's point-to-point communication model with source-channel separation.

In his seminal paper in 1948 [1], Shannon formalized a basic point-to-point communication model along with separate source coding and channel coding theorems, as shown in Fig. 2.1. The information source is generally regarded as a stream of bits, and the sink represents any user of the information. The function of the pair source encoder/decoder, refers to removing redundancy from the source and recovering the original information at the sink, i.e. a process called data compression and restoration. This thesis assumes that the source coding problem has been solved, so that the input to the channel encoder is a sequence of independently identically distributed (i.i.d.) bits which are equally likely to be zero or one. Correspondingly, as a physical medium of transmission, the channel is modelled as a probabilistic mapping "function". Throughout the thesis, we will only

consider the channel coding problem where the channel encoder is employed to protect the information bits from an impairing channel by adding redundant sequences, and the decoder has the task of recovering the original bits, given the received data, despite the existence of noise, different types of signal distortion and interference [7]. The ratio of the number of information bits K to the total number of transmitted bits N is defined as the code rate:

$$R = K/N, \tag{2.1}$$

where $0 < R < 1$. As MacKay asserted in [2], even though one can resort to a physical solution, e.g., using higher power signals, to improve communication reliability, it is more desirable to implement reliable transmission by a system solution, i.e., channel coding. Channel coding theory is concerned with strategies to create practical encoding and decoding systems [2].

2.2.2 Design Criteria and Channel Capacity for Communication Models

Among many channel models, e.g. the binary erasure channel (BEC) or the binary symmetric channel (BSC), the binary-input additive white-Gaussian-noise (BI-AWGN) channel is one of the most frequently used memoryless channels. For this reason, we take the BI-AWGN channel as an example to demonstrate the code design criteria and performance measures. Provided with channel input $\boldsymbol{X} = [x_1, \ldots, x_n]$ and channel output $\boldsymbol{Y} = [y_1, \ldots, y_n]$, for the BI-AWGN channel we map each code bit $c_n \in \{0, 1\}$ to x_n as $x_n = 2(c_n) - 1 \in \{\pm 1\}$, and the discrete-time channel output $y_n = x_n + z_n$ where z_n is a real-valued additive white-Gaussian-noise (AWGN) sample with variance σ^2, i.e. $z_n \sim \mathcal{N}(0, \sigma^2)$. Consequently, the channel can be characterized by the transition probability density function (pdf) $p(y_n|x_n)$ as described by:

$$p(y|x = \pm 1) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-(y \mp 1)^2/(2\sigma^2)], \tag{2.2}$$

and given that $p(x = +1) = p(x = -1) = 1/2$,

$$p(y) = \frac{1}{2}[p(y|x = +1) + p(y|x = -1)]. \tag{2.3}$$

Due to its memorylessness, we have

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_n p(y_n|x_n). \tag{2.4}$$

Apparently, the decoding procedure for the correct codeword is equivalent to minimizing the probability of a codeword error or, equivalently, to maximizing the *a posteriori* probability $p(\boldsymbol{x}|\boldsymbol{y})$. By incorporating Bayes' rule, this MAP rule is given by:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} p(\boldsymbol{x}|\boldsymbol{y}) = \arg\max_{\boldsymbol{c}} \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{2.5}$$

where $\hat{\boldsymbol{c}}$ is an estimate of codeword $\boldsymbol{c}$. We assume that the channel input is equally likely so that either $p(\boldsymbol{x})$ or $p(\boldsymbol{y})$ is independent of $\boldsymbol{c}$. As a result, the MAP rule (2.5) can be replaced by the *maximum-likelihood* (ML) rule [7] as:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} p(\boldsymbol{y}|\boldsymbol{x}). \tag{2.6}$$

However, ML decoding algorithms are very complicated due to their exhaustive nature. Suboptimal decoders, such as iterative decoders, are in favour since they perform slightly worse than ML decoders but are much more efficient [9]. With the word-wise MAP criterion, performance is measured by the word-error rate (WER) or FER as:

$$P_w \triangleq \Pr\{\hat{\boldsymbol{c}} \neq \boldsymbol{c}\}. \tag{2.7}$$

Alternatively, with the bit-wise MAP criterion, the most commonly used performance measure is the BER defined as

$$P_b \triangleq \Pr\{\hat{m}_k \neq m_k\}, \tag{2.8}$$

where m_k denotes the kth information bit and $\hat{m}_k$ denotes its estimate.

Apart from the probabilistic channel model, Shannon also verified that a channel can be characterized by so-called channel capacity C, which measures how much information the channel can convey [1]. From the time of Shannon's seminal work until the early 1990s, it was believed that near capacity codes were incredibly long and impractical for employment in real-life systems. Nevertheless, the invention of turbo codes and the rediscovery of LDPC codes illustrated that capacity-approaching codes are possible in practice [6]. The mutual information between channel input X and channel output Y is defined as $I(X;Y) = H(Y) - H(Y|X)$, where $H(Y)$ is the entropy of Y and $H(Y|X)$ is the conditional entropy of Y given X. Then, the channel capacity is defined as:

$$C = \max_{\{\Pr(x)\}} I(X;Y). \tag{2.9}$$

As long as $R < C$, reliable communication is achievable at code rates R such that R are called achievable rates [1]. The units of capacity C can always be seen as bits of

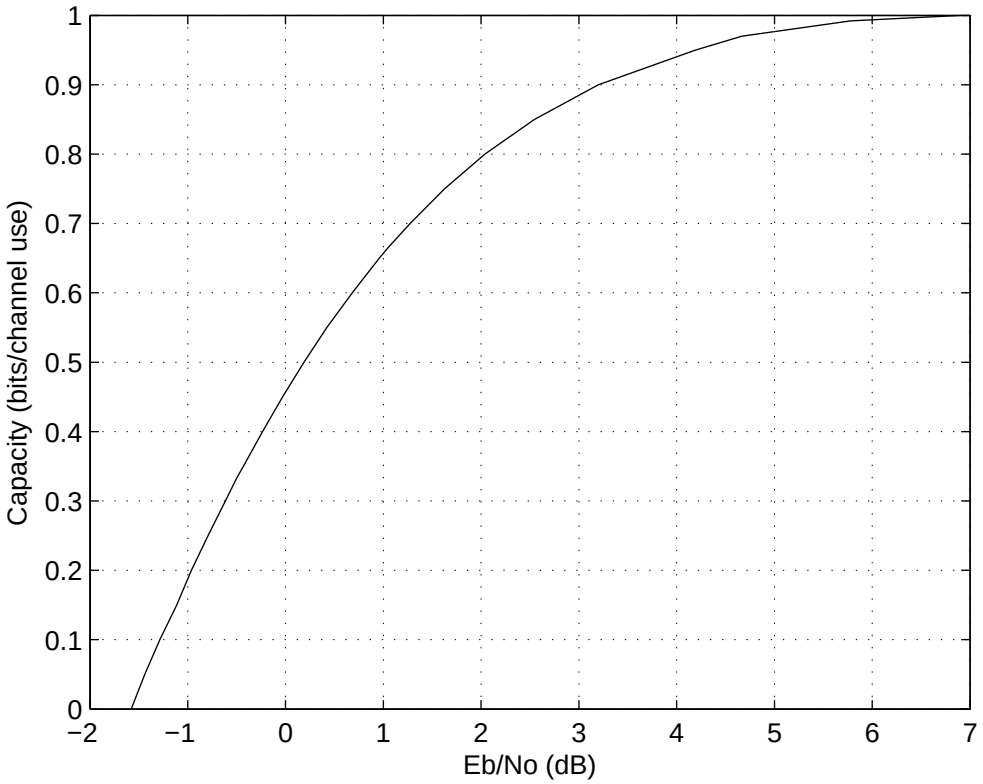

Figure 2.2: Capacity curve for the binary-input AWGN channel.

information. For example, a code rate of 0.5 means that each channel use conveys a half-bit of information. Based on the derivations in [24] - [27], we have the capacity for the BI-AWGN channel:

$$C_{\mathrm{BI-AWGN}} = 0.5 \sum_{x=\pm 1} \int_{-\infty}^{\infty} p(y|x) \log_2 \frac{p(y|x)}{p(y)} dy. \tag{2.10}$$

By using Monte Carlo integration, the integral in (2.10) can be simplified to the expectation $\mathbb{E}\{\cdot\}$. In Fig. 2.2, the channel capacity, C, is plotted against the signal-to-noise-ratio (SNR) denoted by E_b/N_0 in which E_b is the average energy per information bit and $N_0/2 = \sigma^2$ is the two-sided power spectral density of the AWGN that models the channel. Given code rate R, the Shannon capacity, C, also gives a lower limit for SNR in the sense that error-free communication is possible if the SNR just exceeds its limit [7]. Both turbo codes and LDPC codes have been verified as approaching the Shannon limit within 0.5 dB, provided that arbitrarily long code lengths are used [3, 28]. Notice that in this thesis we are mainly concerned with finite-length ($N < 2,000$) practical codes.

2.3 Fundamentals of LDPC Codes

In this section, we introduce the fundamentals of LDPC codes, i.e., parity-check matrices and TGs, along with corresponding classifications. Moreover, two types of algorithms for counting cycles in graphical models, [15] and [16], are presented which play an important role in the following chapters.

2.3.1 Matrix Representation

LDPC codes are a class of linear block codes whose parity-check matrix $\boldsymbol{H}$ has 1% or fewer 1 entries [4]. If only binary codes are considered LDPC codes can be described as a K-dimensional subspace $\mathcal{C}$ of vector space $\mathbb{F}_2^N$ of binary N-tuple over the finite field $\mathbb{F}_2$ [7]. Given a $K \times N$ generator matrix $\boldsymbol{G}$, a codeword $\boldsymbol{c} \in \mathcal{C}$ is obtained by:

$$\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G}, \tag{2.11}$$

where $\boldsymbol{m}$ is a binary row vector containing K bits. Accordingly, an $(N-K) \times N$ parity-check matrix $\boldsymbol{H}$ forms the null space $\mathcal{C}^{\perp}$ so that $\boldsymbol{H}\boldsymbol{G}^T = \boldsymbol{0}$. The matrix $\boldsymbol{H}$ is so named since it consists of $N - K = M$ separate parity-check equations as described by:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \longmapsto \begin{pmatrix} c_1 \bigoplus c_2 \bigoplus c_3 = 0 \\ c_3 \bigoplus c_4 \bigoplus c_5 = 0 \\ c_4 \bigoplus c_5 \bigoplus c_6 = 0 \end{pmatrix}, \tag{2.12}$$

in which c_n denotes the nth code bit and "$\bigoplus$" is the modulo 2 addition (mod 2). Based on the above description, the fundamental condition for error correction which is sought by the decoder is given by:

$$\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{0}, \tag{2.13}$$

which is often used as a criterion to determine whether the decoder finds the correct codeword ($\hat{\boldsymbol{c}} = \boldsymbol{c}$). Unlike the example in (2.12), practical LDPC codes have a key feature, i.e. "low density", which allows efficient iterative decoding algorithms [10]. As we know, optimum (e.g. ML) decoding for normal linear block codes is not possible because of its prohibitive complexity [9]. Nevertheless, thanks to their low-density feature, LDPC codes fully take advantage of iterative decoding, which can achieve near-optimal performance

at low error rates for many applications [7]. With a parity-check matrix $\boldsymbol{H}$ having column weight w_c and row weight w_r, an LDPC code is called regular when both w_c and w_r are constants, otherwise the code is called irregular. For regular codes, the code rate is:

$$R = 1 - \frac{M}{N} = 1 - \frac{w_c}{w_r}. \tag{2.14}$$

In most designs, we need to transform $\boldsymbol{H}$ into $\boldsymbol{G}$ by using a transition matrix $\boldsymbol{P}$ such that $\boldsymbol{H} = [\boldsymbol{I}|\boldsymbol{P}]$ and $\boldsymbol{G} = [\boldsymbol{P}^T|\boldsymbol{I}]$. To implement the transformation and validate equation (2.14), the matrix $\boldsymbol{H}$ has to be full rank. If not, a process of Gaussian Elimination (GE) is required.

2.3.2 Graphical Representation

In his pioneering work [12], Tanner studied LDPC codes and illustrated how they can be represented by the so-called Tanner graph, or TG for brevity, which is similar to the trellis graph of a convolutional code in the sense of facilitating description of the code and relevant algorithms. A TG is a bipartite graph whose nodes are separated into two categories, i.e. variables nodes (or symbol nodes) and check nodes (or constraint nodes), respectively. Each TG uniquely corresponds to a parity-check matrix: there is an edge connecting check node C_i and variable node V_j in the factor graph only if entry h_{ij} of the parity-check matrix $\boldsymbol{H}$ equals 1. For instance, the TG for $\boldsymbol{H}$ in (2.12) is shown in Fig. 2.3. Observe from Fig. 2.3 that M check nodes specify M parity-check equations of $\boldsymbol{H}$,

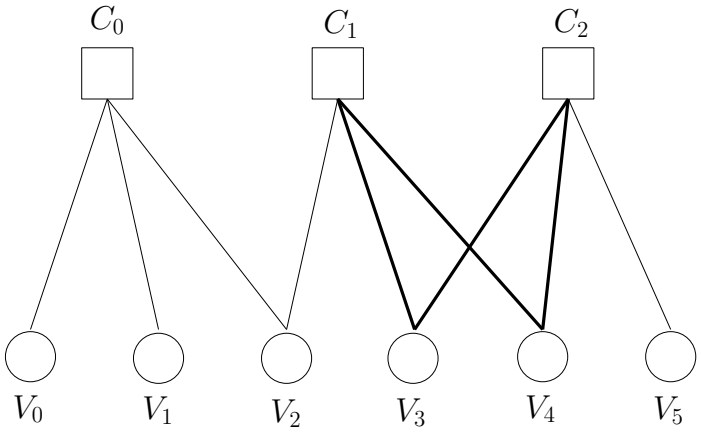

Figure 2.3: The TG for the parity-check matrix $\boldsymbol{H}$ given in (2.12).

and N variable nodes represent N code bits in the code. As for the code design, more

connections between variable nodes and check nodes would provide more protection to each code bit, but the sparseness (low density) of the $\boldsymbol{H}$ matrix is a characteristic that facilitates the use of iterative decoding [4].

From a TG perspective, regular codes have uniform column weight and row weight while w_c and w_r for irregular codes may vary with columns and rows. It is convenient to elaborate those parameters by using degree distribution polynomials, denoted by $\upsilon(x)$ and $\nu(x)$, respectively, as:

$$\upsilon(x) = \sum_{d=1}^{d_v} \upsilon_d x^{d-1}, \quad \nu(x) = \sum_{d=1}^{d_c} \nu_d x^{d-1}, \tag{2.15}$$

where υ_i is the fraction of edges connected to degree d check nodes, ν_j is the fraction of edges connected to degree d variable nodes, d_v and d_c denote the maximum variable node degree and the maximum check node degree, respectively. Notice that the polynomials $\upsilon(x) = x^{d_v-1}$ and $\nu(x) = x^{d_c-1}$ correspond to regular codes. Let us further denote by E the total number of edges (connections) in the TG, then:

$$E = \frac{N}{\int_0^1 \upsilon(x)dx} = \frac{M}{\int_0^1 \nu(x)dx}. \tag{2.16}$$

In Fig. 2.3, notice that four thickened edges form a closed loop which is usually referred to as a cycle in the literature. As observed in [4], the performance of an iterative decoder depends on the structural properties of the TG on which the decoding is based. In particular, the presence of short cycles considerably diminishes the effectiveness of iterative decoding algorithms applied to LDPC codes, since the decoder works locally, around short cycles, so that the exchange of decoding information only takes place in some portions of the TG without a globally optimum solution, [29] and [30]. On the other hand, short cycles are inevitable, especially for LDPC codes with finite blocks, [14] and [21]. Consequently, how to avoid short cycles of certain lengths and how to mitigate the effect of those cycles become central issues in this thesis.

2.3.3 Effective Algorithms for Counting Short Cycles

Following the rediscovery of LDPC codes, [4] and [31], work by Wiberg, Loeliger and Kötter, [32] and [33], stimulated interest in codes based on graphical models. The mini-

mum length of cycles is often defined as g, the girth of a graph. In [34] - [37], it is widely accepted that good graphical codes have large girth and a small number of short cycles in their graphical counterparts, such as TG or other factor graphs. With the advent of irregular LDPC codes [28], the cycle structure and its regularity became another design criterion to have good graphical models for codes. To have an in-depth understanding of the graph structure or cycle distribution, we need effective algorithms that can count cycles of distinct length in the graph. However, counting short cycles exactly in an arbitrary graph seems to be a difficult problem. In [38], Alon et al. proposed methods to count short cycles but the complexity is prohibitively high for cycles with length over 7. A similar issue also arose with algorithm [39] for counting cycles in the TG of LDPC codes.

One of the most efficient algorithms for counting cycles was presented by Halford and Chugg [15]. This algorithm successfully transforms the problem of counting cycles into one of counting so-called lollipop walks since counting all the short cycles exactly in a bipartite graph is seen to be computationally infeasible. The term $(m, n-m)$-lollipop walk refers to length n walks, i.e. a sequence of vertices $\{a_1, \ldots, a_{n+1}\}$, where all vertices are distinct except $a_{n+1} = a_{m+1}$. Thus, cycles of length $2m$ are $(0, 2m)$-lollipop walks. We consider a bipartite graph $\mathcal{G}(V, E)$ where the set V consists of two separate sets of nodes V_c and V_s and the set E denotes the edge set. Given that '$|\cdot|$' represents the cardinality of a set, let us define $\boldsymbol{P}_{2k}^{v_c}$ as a $\mid V_c \mid \times \mid V_c \mid$ matrix in which the $(i,j)th$ element is the number of paths of length $2k$ from $v_{c_i} \in V_c$ to $v_{c_j} \in V_c$. Similarly, let us define $\boldsymbol{P}_{2k+1}^{v_c}$ as a $\mid V_c \mid \times \mid V_s \mid$ matrix in which the $(i,j)th$ element is the number of paths of length $2k+1$ from $v_{c_i} \in V_c$ to $v_{s_j} \in V_s$. Also let us define $\boldsymbol{L}_{2k',2k-2k'}^{v_c}$ as a $\mid V_c \mid \times \mid V_c \mid$ matrix in which the $(i,j)th$ element is the number of $(2k', 2k-2k')$-lollipop walks from $v_{c_i} \in V_c$ to $v_{c_j} \in V_c$. Likewise, let us define $\boldsymbol{L}_{2k'+1,2k-2k'}^{v_c}$ as a $\mid V_c \mid \times \mid V_s \mid$ matrix in which the $(i,j)th$ element is the number of $(2k'+1, 2k-2k')$-lollipop walks from $v_{c_i} \in V_c$ to $v_{s_j} \in V_s$. The afore mentioned matrices satisfy the following relationships:

$$\boldsymbol{P}_{2k+1}^{v_c} = \boldsymbol{P}_{2k}^{v_c}\boldsymbol{E} - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i+1,2k-2i)}^{v_c}, \tag{2.17}$$

$$\boldsymbol{P}_{2k}^{v_c} = \boldsymbol{P}_{2k-1}^{v_c}\boldsymbol{E}^T - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i,2k-2i)}^{v_c}, \tag{2.18}$$

$$\boldsymbol{P}_{2k+1}^{v_s} = \boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E}^T - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i+1,2k-2i)}^{v_s}, \tag{2.19}$$

$$\boldsymbol{P}_{2k}^{v_s} = \boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E} - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i,2k-2i)}^{v_s}, \tag{2.20}$$

$$\boldsymbol{L}_{(0,2k)}^{v_c} = (\boldsymbol{P}_{2k-1}^{v_c}\boldsymbol{E}^T) \circ \boldsymbol{I}, \tag{2.21}$$

$$\boldsymbol{L}_{(0,2k)}^{v_s} = (\boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E}) \circ \boldsymbol{I}, \tag{2.22}$$

where '$\circ$' means element-wise matrix multiplication, $\boldsymbol{E}$ is an edge matrix whose $(i,j)th$ entry equals 1 if there is a connection between the ith and the jth nodes, and $\boldsymbol{I}$ is an identity matrix. The total number of cycles of length $2k$ is given by:

$$N_{2k} = \frac{1}{2k}\mathrm{Tr}\big(\boldsymbol{L}_{(0,2k)}^{v_c}\big) = \frac{1}{2k}\mathrm{Tr}\big(\boldsymbol{L}_{(0,2k)}^{v_s}\big), \tag{2.23}$$

where '$\mathrm{Tr}(\cdot)$' means the trace of the matrix in the argument. In order to find the girth g and count cycles of length g, $g+2$ and $g+4$ in a Tanner Graph, (2.17) - (2.22) are expanded and updated with each other such that counting short cycles is equal to counting lollipop recursions. Apart from the one described above, another efficient algorithm for counting cycles was recently proposed by Karimi et al. in [16], which takes advantage of message passing computation rather than matrix multiplication. For bipartite graphs, the first algorithm is able to count cycles of length g, $g+2$, $g+4$ with complexity of $\mathcal{O}(gN^3)$, while the second algorithm can count cycles of length g, $g+2$, $\ldots$, $2g-2$ with a lower complexity, $\mathcal{O}(g|E|^2)$ ($|E|$ is the total number of edges in the graph). Yet, the second algorithm implicitly assumes that the girth is known *a priori* so that greater complexity for $\mathcal{O}(N^2)$ may be needed to find the girth [16].

2.4 Construction of LDPC Codes

From the content in Section 2.3.1, it is obvious that the key constructing of an LDPC code is to construct its low-density parity-check matrix with some desired properties. There are a great number of design methods in the literature. Generally speaking, design techniques to construct parity-check matrices of LDPC codes fall into two main categories: computer-based and algebraic methods. The algebraic approach often involves finite mathematics, [11], [40], and [41], or combinatorial techniques, [42] - [45], which are

promising for industrial applications thanks to simple encoding structures. On the other hand, computer-based techniques, including Gallager codes [10], MacKay codes [4] and density evolution (DE), [28] and [46], are still predominant as those random constructions are highly flexible in their code design and can offer near-capacity performance with very large block lengths. In this section, we will present some of the most important computer-based techniques.

2.4.1 Gallager Codes

In his original work, [10], Gallager first proposed regular LDPC codes with three parameters (N, w_c, w_r) to denote the code length, the number of 1s in each column, and the number of 1s in each row, respectively. A parity-check matrix $\boldsymbol{H}$ for Gallager codes is constructed by random column permutations, and has the following structure:

- The parity-check matrix $\boldsymbol{H}$ can be split into w_c submatrices $\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{w_c}$. Each submatrix consists of $\lfloor (N - w_r)/w_c \rfloor$ rows and has only one element equal to 1 in every column.
- For $\boldsymbol{H}_1$, the row elements equal to 1 are arranged in sloping fashion, i.e. all w_r elements of 1 in the ith row $(1 \leq i \leq w_r)$ should be placed from $(i-1)w_r + 1$ to iw_r.
- The rest of the submatrices $\boldsymbol{H}_2, \ldots, \boldsymbol{H}_{w_c}$ are produced via column permutations of $\boldsymbol{H}_1$.

If the parity-check matrix of an (N, w_c, w_r) regular LDPC code is full rank, and provided that $\boldsymbol{H}$ has M rows, then the total number of elements equal to 1 in $\boldsymbol{H}$ is:

$$M \times w_r = N \times w_c \Rightarrow \frac{M}{N} = \frac{w_c}{w_r}. \tag{2.24}$$

As an example, the parity-check matrix $\boldsymbol{H}$ for a Gallager code $(20, 3, 4)$ is shown below

$$\boldsymbol{H} = \left[\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hdashline
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hdashline
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\right] \tag{2.25}$$

After being ignored for decades, Gallager codes were generalized by Tanner in 1981 [12], then further studied by MacKay and others, [4] and [5].

2.4.2 MacKay Codes

An alternative construction method for LDPC codes was invented by MacKay [4] while presumably not being aware of Gallager codes. The approach illustrates the benefits of designing codes with sparse $\boldsymbol{H}$ matrices, and for the first time shows the capability of LDPC codes to perform near capacity limits [5]. On the basis of TG, MacKay codes imposes an important structural property on $\boldsymbol{H}$: no two rows or two columns have more than one position in common that contains a 1 element, that is referred to as the row-column constraint [7]. During iterative decoding, if two variable nodes participate in two parity-check equations and two parity-check equations corrupt simultaneously, it is not possible to detect the corrupted bits and correct them. The row-column constraint eliminates short cycles of length 4 in TG since the presence of such cycles significantly degrades the performance of iterative decoding algorithms.

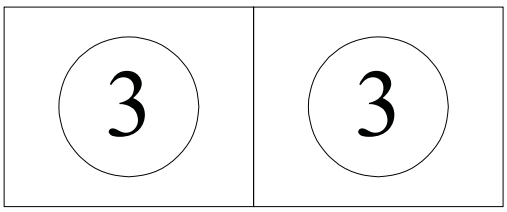

Figure 2.4: MacKay C-1A construction whose parity-check matrix consists of two different sub-matrices with column weight 3.

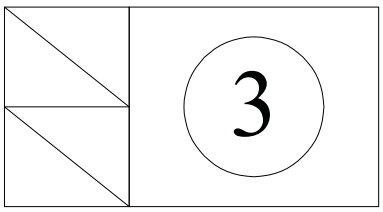

Figure 2.5: MacKay C-2A construction whose parity-check matrix is comprised of a sub-matrix with column weight 3 as well as two identity matrices.

Fig. 2.4 [4] depicts one of the structures of Mackay codes. The MacKay C-1A construction ensures that every column in the parity-check matrix $\boldsymbol{H}$ only has t elements of 1 and randomly sets their positions in the matrix, but the number of 1s in every single row is distributed as uniformly as possible. The row-column constraint also applies to the MacKay C-1A construction. To retain the nature of sparseness, it is required to reduce the number of 1s in $\boldsymbol{H}$ by introducing some low-weight columns, e.g. $t = 2$. However, the probability of low-weight columns needs to be small, as too many low- degree columns will lead to some information bits lacking enough protection. Based on this observation, the basic MacKay C-1A construction can be naturally extended to the Mackay C-2A construction, as illustrated in Fig. 2.5. Containing a total of M columns, the Mackay C-2A construction generates a parity-check matrix which has $M/2$ columns whose column weight t is quite low (say $t = 2$). Two $M/2 \times M/2$ identity matrices are located on the left-hand side from top to bottom, while the other $N - M/2$ columns are filled in the same way as in the MacKay C-1A.

2.4.3 Density Evolution and EXIT Charts

To improve the iterative decoding performance in the "waterfall" region, Richardson et al. [28] and Luby et al. [46] designed ensembles of irregular LDPC codes defined by the pair of degree distributions in (2.15) and showed how to optimize those distribution polynomials for a number of channels. In [28] and [47], the authors assert that the the performance of very long codes is predictable via the ensemble average performance, which is equivalent to the performance under the assumption of being cycle free. Moreover, an

algorithm called density evolution (DE) was proposed which refers to the evolution of the probability density functions (pdfs) of the messages being passed during iterative decoding. With knowledge of the pdfs, one can predict in what channel conditions (e.g., SNRs) the decoding of BER will converge to zero. The error probability depends on the channel parameter, e.g. the standard deviation σ of channel noise for the AWGN channels. In [47], it is proven that long LDPC codes have a decoding threshold σ^* such that if $\sigma < \sigma^*$ then $Pr(\text{error}) \to 0$ as the number of iterations $l \to \infty$; otherwise, $Pr(\text{error})$ will be bound away from zero. In other words, the DE algorithm allows us to determine the decoding threshold of an LDPC code ensemble provided with block length $N \to \infty$ and unlimited decoding iterations [7].

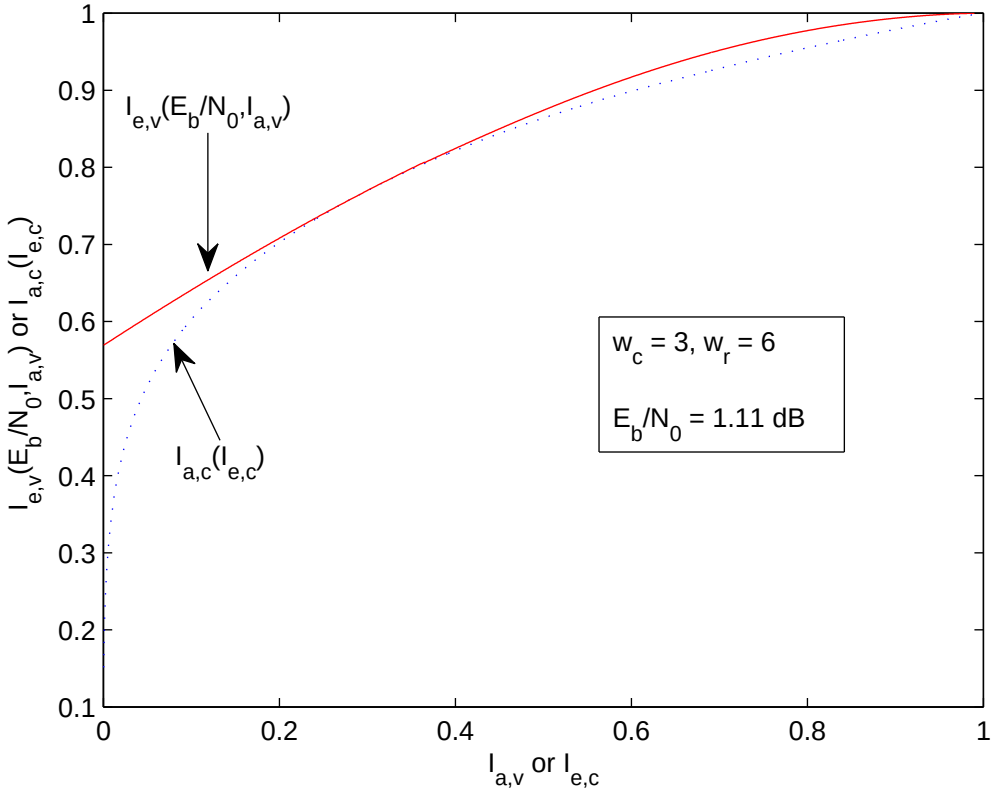

Figure 2.6: EXIT chart for regular LDPC $(w_c = 3, w_r = 6)$ code ensemble.

As an alternative approach to DE, the extrinsic-information-transfer (EXIT) chart is a graphical technique for predicting the decoding thresholds of LDPC code ensembles. Covered in the literature, [48] - [51], EXIT charts are inspired by the fact that the variable nodes and check nodes work cooperatively and iteratively to make decoding decisions, with extrinsic mutual information consistently improving with iterations. Such a chart

helps to estimate the decoding threshold of the ensemble of LDPC codes represented by the degree of distribution of variable nodes and check nodes. Similar to DE, the decoding threshold predicted by EXIT charts is subject to a cycle-free TG, an infinite code length, and an infinite number of decoding iterations, [6]. In Fig. 2.6, the EXIT chart is plotted for the ensemble of regular LDPC codes with column weight $w_c = 3$ and row weight $w_r = 6$ over the BI-AWGN channel. The red (solid) curve depicts the mutual information, $I_{e,v}$, for the extrinsic information coming out of the variable nodes against the mutual information, $I_{a,v}$, for the extrinsic information (*a priori*) going into the variable nodes. Likewise, the blue (dashed) curve depicts the mutual information, $I_{a,c}$, for the extrinsic information (*a priori*) information going into the check nodes against the mutual information, $I_{e,c}$, for the extrinsic information coming out of the check nodes. When the channel SNR increases ($E_b/N_0 > 1.1dB$), the red curve (variable nodes) moves upwards, which increases the "tunnel" between the two curves as well as the decoding convergence rate. If the SNR $E_b/N_0 < 1.1dB$, the tunnel will be closed such that the two curves cannot meet at $(I_{e,v}, I_{e,c}) = (1, 1)$, point for which the error rate is zero.

Both DE and EXIT chart techniques are effective tools for the design and asymptotic analysis of LDPC codes. Nevertheless, the degree distribution pairs are only optimal in the sense that the block length is very long ($N > 5,000$) and the rate is not too high ($R \leq 3/4$) [52], [53]. In [54], the authors state that $\upsilon(x)$-$\nu(x)$ pairs are suboptimal for medium or short codes and give rise to a high error floor.

2.4.4 Progressive Edge-Growth (PEG) Algorithm

As mentioned previously, short cycles in the TG hinder successful decoding for iterative decoders. Among the many algorithms dealing with girth conditioning, the PEG algorithm [21, 22] is one of the most effective methods for computer-based code design. In this thesis, the main focus is on computer-based design techniques that are similar to the interleavers of turbo codes, where the randomness or pseudo-randomness lying within computer-based constructions significantly enhances the overall performance of codes [6]. Provided with N, M and degree distributions, the PEG algorithm builds the TG in edge-by-edge fashion such that every edge added maximizes the girth g as well as minimizing the number of short cycles. From level-0 to level-d_{max}, the algorithm creates

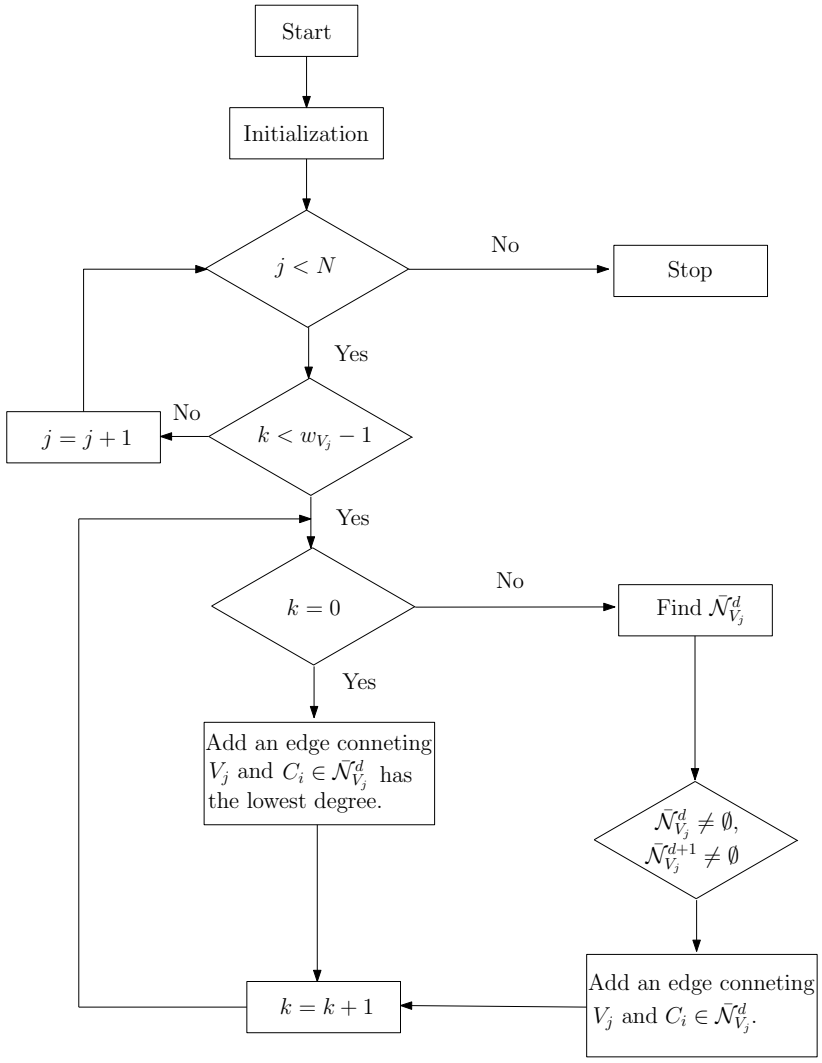

Figure 2.7: The algorithm flowchart of the PEG LDPC code design.

the TG from a root variable node V_j with degree w_{V_j}. Edge placement always starts with the lowest-degree variable nodes since they receive the least amount of help from their neighbourhood $\mathcal{N}_{V_j}$ and are the most vulnerable to error [21]. Likewise, the first edge connecting V_j is placed with the lowest-degree check node C_i to yield a fairly uniform degree distribution of check nodes. Following the flowchart depicted in Fig. 2.7, a new edge to V_j will be attached to an unreachable check node such that no cycle is created. Otherwise, if all check nodes in $\bar{\mathcal{N}}_{V_j}^d$, the complementary set of $\mathcal{N}_{V_j}$ at level-d, are reachable, an edge is created to connect V_j to the most distant check node (resulting in the largest girth).

The BER and FER performance comparisons of the Gallager design, MacKay design and PEG algorithm are shown in Figs. 2.8 and 2.9, respectively. For all three code

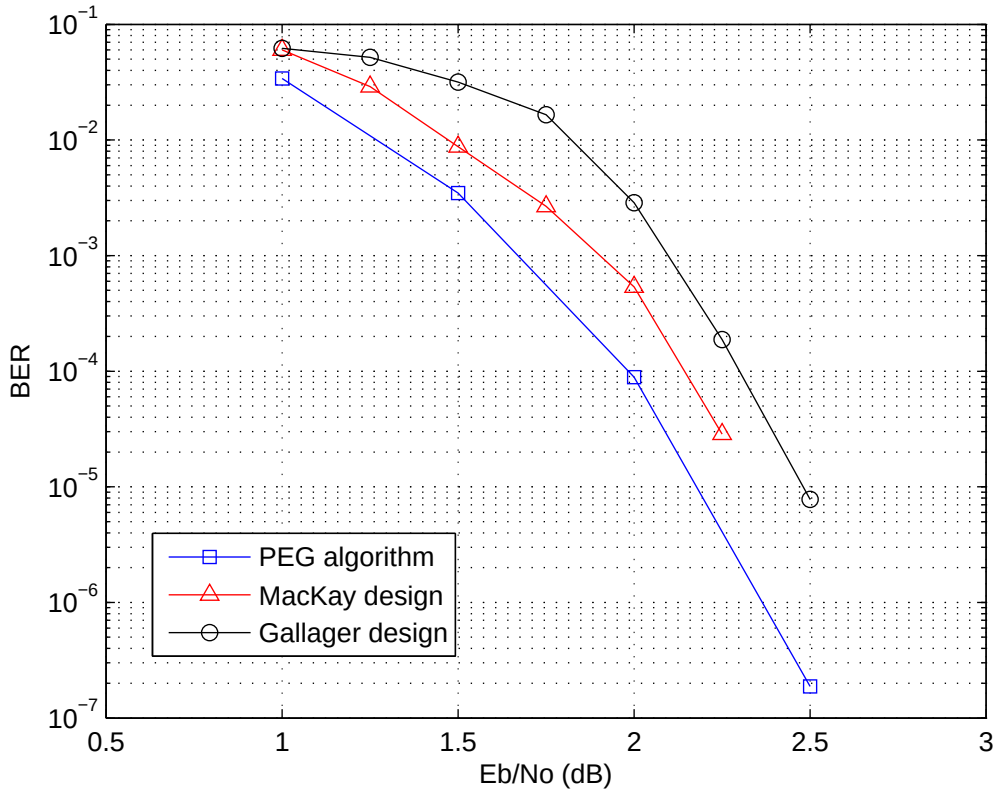

Figure 2.8: Comparison of the BER performances of Gallager codes, MacKay codes and the PEG algorithm with a maximum number of 60 decoding iterations.

designs, block size $N = 2000$, code rate $R = 1/2$ and the maximum number of decoding iterations is 60. The Gallager code has $w_c = 3$ and $w_r = 6$ while the MacKay and PEG codes are irregularly designed with $w_c = 2, 3, 4$ and $w_r = 6$. Note that the PEG algorithm is theoretically applicable to any block length but a large block size, say $10,000$ may lead to a very slow design process.

2.5 Iterative Message-Passing Algorithms for Decoding LDPC Codes

BP algorithm, also known as sum-product algorithm (SPA), is a powerful algorithm to solve inference problems in statistical physics, computer vision and error control coding

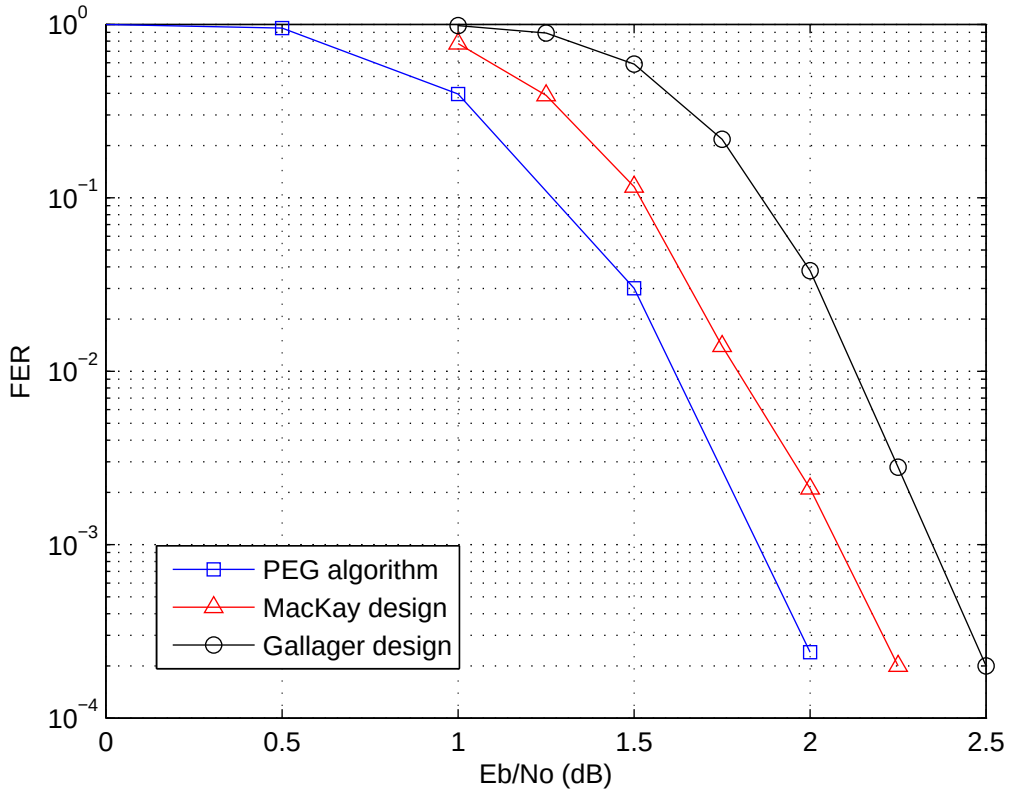

Figure 2.9: Comparison of the FER performances of Gallager codes, MacKay codes and the PEG algorithm with a maximum number of 60 decoding iterations.

approximately [55]. This message passing algorithm aims to derive accurate marginal distributions of variables corresponding to each node of a graphical model, and is exceptionally useful when optimal inference decoding is computationally prohibitive due to the substantial size of a graph [56]. Since being applied to decoding turbo and LDPC codes, there have been various versions of BP graph-based decoding algorithms reported in the area, [3], [28], [57] and [81]. Although the BP algorithm is capable of producing exact inference solution only if the graphical model is acyclic (i.e. a tree), the decoding performance for both turbo codes and LDPC codes is still very impressive in practice [13].

2.5.1 Probabity-Domain BP decoder

The first BP decoding algorithm dates back to 1960 when Gallager invented LDPC codes for his PhD thesis at MIT [59]. This algorithm works in the probability domain, and aims to find the most probable vector $\boldsymbol{x}$ such that $\boldsymbol{H}\boldsymbol{x}^T = \mathbf{0}$. Prior to the description of standard BP algorithms, basic notation is given as follows:

$$f_n^x = Pr(x_n = x) \quad and \quad f_n^0 + f_n^1 = 1, \tag{2.26}$$

$N(m) = \{n : H_{mn} = 1\}$ denotes the set of variable nodes connecting to the check node m where $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$. Similarly, $M(n) = \{m : H_{mn} = 1\}$ denotes the set of check nodes connecting to the variable node n, i.e. the positions of 1 entries of $\boldsymbol{H}$ in the nth column. In addition, $N(m)\backslash n$ represents the neighbouring set of check node m except for the variable node n. Likewise, $M(n)\backslash m$ is the neighbouring set of variable node n except for the check node m. q_{mn}^x denotes the probability that variable node n equals x given the messages sent from the neighbours of n except m. The quantity r_{mn}^x denotes the probability that the parity-check equation m is satisfied while n equals x along with the probability distribution of other neighbours of m. In summary, the standard BP algorithm keeps updating two probabilities q_{mn} and r_{mn} in an iterative fashion. In the case of a cycle-free TG, it is guaranteed to calculate precisely the *a posteriori* probability of each code bit [57]. The complete process of BP decoding is described as follows:

1. Initialization

   For all check nodes $m \in 1, 2, \ldots, M$ and variable nodes $n \in 1, 2, \ldots, N$, q_{mn}^0 and q_{mn}^1 are initialized as f_n^0 and f_n^1, respectively.

2. Iterative decoding starts:

   (a) Define $\delta_{q_{mn}} = q_{mn}^0 - q_{mn}^1$ for all m and n where $h_{mn} = 1$, then compute:

   $$\delta_{r_{mn}} = \prod_{n' \in N(m)\ n} \delta_{q_{mn'}}, \tag{2.27}$$

   $$r_{mn}^x = \frac{1}{2}\left(1 + (-1)\delta_{q_{mn}}\delta_{r_{mn}}\right). \tag{2.28}$$

   (b) Assuming $x \in \{0, 1\}$ update

   $$q_{mn}^x = \alpha_{mn} f_n^x \prod_{m' \in N(n)\ m} \delta_{r_{m'n}}, \tag{2.29}$$

where α_{mn} functions as a constraint factor such that $q^0_{mn} + q^1_{mn} = 1$. Meanwhile, update the *a posteriori* probability q^x_n using

$$q^x_n = \alpha_n f^x_n \prod_{m' \in N(n)\ m} \delta_{r_{m'n}}, \tag{2.30}$$

where α_n is another constraint factor that ensures $q^0_n + q^1_n = 1$.

(c) Make decisions such that $\hat{x_n} = 1$ if $q^1_n > 0.5$ or $\hat{x_n} = 0$ if $q^1_n < 0.5$. Iterative decoding stops in the case of $\boldsymbol{H}^T\hat{\boldsymbol{x}} = 0$, then $\hat{\boldsymbol{x}}$ is the decoded codeword. Otherwise return to (a). If the correct codeword has not been found when the maximum number of iterations is reached, the algorithm declares a decoding failure or outputs $\hat{\boldsymbol{x}}$ as the decoded codeword.

The computational complexity of the BP algorithm is directly related to the number of non-zero entries in the parity-check matrix $\boldsymbol{H}$. This feature enables efficient application when decoding LDPC codes [2].

2.5.2 Log-Domain BP decoder

Alternatively, for numerical stability, messages are often calculated in the logarithmic domain by replacing probabilities by log-likelihood ratios (LLRs) as:

$$L(x) = \ln \frac{Pr(x=0)}{Pr(x=1)}. \tag{2.31}$$

With LLR representation, gains in computation and storage can be achieved, [7] and [9]. To describe the log-domain BP decoder, the following definitions are given: $L_{(n)} = \ln \frac{f^0_n}{f^1_n}$, $\lambda_{mn} = \ln \frac{q^0_{mn}}{q^1_{mn}}$, $\lambda_n = \ln \frac{q^0_n}{q^1_n}$, and $\Lambda_{mn} = \ln \frac{r^0_{mn}}{r^1_{mn}}$. Moreover, based on (2.27) and (2.28), we

can derive:

$$\begin{aligned}\Lambda_{mn} &= \ln \frac{1+\prod_{n'\in N(m)\ n}(q^0_{mn'}-q^1_{mn'})}{1-\prod_{n'\in N(m)\ n}(q^0_{mn'}-q^1_{mn'})} \\ &= 2\tanh^{-1}\left(\prod_{n'\in N(m)\ n}(q^0_{mn'}-q^1_{mn'})\right) \\ &= 2\tanh^{-1}\frac{\prod_{n'\in N(m)\ n}(\frac{q^0_{mn'}}{q^1_{mn'}}-1)}{\prod_{n'\in N(m)\ n}(\frac{q^0_{mn'}}{q^1_{mn'}}+1)} \\ &= 2\tanh^{-1}\left(\prod_{n'\in N(m)\ n}(\frac{e^{\lambda_{mn'}}-1}{\lambda_{mn'}+1}\right) \\ &= 2\tanh^{-1}\left(\prod_{n'\in N(m)\ n}\tanh\frac{\lambda_{mn'}}{2}\right),\end{aligned} \tag{2.32}$$

and

$$\begin{aligned}\lambda_{mn} &= \ln\frac{q^0_{mn}}{q^1_{mn}} \\ &= \ln\frac{\alpha_{mn}f^0_n\prod_{m'\in N(n)\ m}r^0_{m'n}}{\alpha_{mn}f^1_n\prod_{m'\in N(n)\ m}r^1_{m'n}} \\ &= \ln\frac{f^0_n}{f^1_n}+\sum_{m'\in N(n)\ m}\ln\frac{r^0_{m'n}}{r^1_{m'n}} \\ &= L_n+\sum_{m'\in N(n)\ m}\Lambda_{m'n},\end{aligned} \tag{2.33}$$

Obviously, we then have:

$$\lambda_n = L_n+\sum_{m\in N(n)}\Lambda_{mn}. \tag{2.34}$$

Provided with (2.32) - (2.34), the BP decoding algorithm in the log-domain can be described as follows:

Step 1: Initialization

For all variable nodes n and check nodes $m\in N(n)$, compute:

$$\lambda_{mn} = L_n, \tag{2.35}$$

where L_n represents the channel characteristics. Given that BI-AWGN channels with 0 and 1 are equally likely to be transmitted, assume the received signal is y_n, then:

$$\begin{aligned}p(y_n|x_n=+1) &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y_n-1)^2}{2\sigma^2}} \\ p(y_n|x_n=-1) &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y_n+1)^2}{2\sigma^2}}\end{aligned} \tag{2.36}$$

By applying Bayes' rule we obtain:

$$\frac{p(x_n = -1|y_n)}{p(x_n = +1|y_n)} = \frac{p(y_n|x_n = -1)}{p(y_n|x_n = +1)}, \tag{2.37}$$

Given (2.36) and (2.37), for BI-AWGN channels we have

$$L_n = -\frac{2y_n}{\sigma^2}. \tag{2.38}$$

Step 2: Update messages from check nodes

For all check nodes m and variable nodes $n \in N(m)$, calculate

$$\Lambda_{mn} = 2\tanh^{-1}\left(\prod_{n' \in N(m)\ n} \tanh\frac{\lambda_{mn'}}{2}\right). \tag{2.39}$$

Step 3: Update messages from variable nodes

For all variable nodes n and check nodes $m \in N(n)$, compute

$$\lambda_{mn} = L_n + \sum_{m' \in N(n)\ m} \Lambda_{m'n}, \tag{2.40}$$

Then update λ_n as

$$\lambda_n = L_n + \sum_{m \in N(n)} \Lambda_{mn}. \tag{2.41}$$

Step 4: Make decoding decisions

$\hat{x_n} = 1$ if $\lambda_n > 0$, otherwise $\hat{x_n} = 0$ if $\lambda_n < 0$. Similar to the BP algorithm in the probability domain, the decoding terminates if the condition $\boldsymbol{H}^T\hat{\boldsymbol{x}} = 0$ is satisfied or the maximum number of iterations is completed. The importance of the log-domain BP decoder lies in its being capable of considerably reducing both complexity and storage. Most importantly, this is not accomplished by sacrificing the decoding performance [13].

2.5.3 Reduced Complexity Decoders

In [60], the authors presents a number of algorithms which can further simplify the procedure of the standard BP algorithm, even if they reduce complexity at the cost of reducing performance. Now we briefly review one of the simplified decoding algorithms, i.e., the min-sum algorithm. To reduce the computational complexity in (2.39), the hyperbolic functions $\tanh(\cdot)$ and $\tanh^{-1}(\cdot)$ can be simplified by using appropriate approximations. Given two random variables U and V, we define:

$$L(U \oplus V) = 2\tanh^{-1}\left(\tanh\left(\frac{L(U)}{2}\right)\tanh\left(\frac{L(V)}{2}\right)\right), \tag{2.42}$$

and it can be expressed as:

$$L(U \oplus V) = \log\left(\frac{1 + e^{L(U)+L(V)}}{e^{L(U)} + e^{L(V)}}\right). \tag{2.43}$$

Now, using the Jacobian algorithm [13] and [92] for approximation, we obtain:

$$\begin{aligned} L(U \oplus V) =& \text{sign}\left(L(U)\right) \text{sign}\left(L(V)\right) \min\left(|L(U)|, |L(V)|\right) \\ &+ \log\left(1 + e^{-|L(U)+L(V)|}\right) - \log\left(1 + e^{-|L(U)-L(V)|}\right) \\ &\approx \text{sign}\left(L(U)\right) \text{sign}\left(L(V)\right) \min\left(|L(U)|, |L(V)|\right). \end{aligned} \tag{2.44}$$

Eventually, the log-domain BP algorithm is transformed into the min-sum algorithm by replacing (2.39) in Step 2 with:

$$\Lambda_{mn} = \prod_{n' \in N(m)\ n} \text{sign}(\lambda_{mn'}) \min(|\lambda_{mn'}|). \tag{2.45}$$

Compared to the standard BP algorithm, the min-sum algorithm is well known for its low complexity thanks to only involving additions and min-operations. Nevertheless, the decoding performance of the min-sum decoder is significantly worse than that of the standard BP decoder. As a result, there is a trade-off between complexity and performance with regards to various LDPC decoding algorithms.

Chapter 3

Finite-Length Rate-Compatible LDPC Codes Based on Puncturing and Extension Techniques

Contents

3.1 Overview

In this chapter, we investigate novel strategies for generating rate-compatible (RC) irregular low-density parity-check (LDPC) codes with short/moderate block lengths. We propose three puncturing and two extension schemes, which are designed to determine the

puncturing positions that minimize performance degradation. The first puncturing scheme employs a cycle-counting algorithm and a grouping strategy for variable nodes having short cycles of equal length in the TG. The second scheme relies on a metric called EMD and the third scheme is a simulation-based exhaustive search to find the best puncturing pattern among several random ones. In addition, we devise two layer-structured extension schemes based on a counting cycle algorithm and an EMD metric which are applied to design RC-LDPC codes. An analysis is provided to explain how the proposed puncturing schemes are able to determine puncturing patterns which minimize performance degradation. Simulation results show that the proposed extension and puncturing techniques achieve better rate flexibility and minimize the performance loss over the AWGN channel, outperforming existing techniques.

3.2 Introduction

When the channel state information (CSI) is known at the transmit end and the data transmission takes place over time-varying channels, an error control coding scheme with a fixed code rate is not regarded as the best solution. In such a situation, an error-correction scheme with flexibility in code rates is desirable since it is able to encode data at different rates depending on the reliability of the channel. Higher rate codes are applied to achieve higher data throughput if the channel condition is good, otherwise lower rate codes are used to guarantee reliable transmission. Thus, both capacity and reliability can be realized in such a scenario. However, deploying many pairs of encoders and decoders is not feasible in practical applications due to their high cost. RC codes refer to a family of codes where higher rate codes are embedded in lower rate codes; in other words, the factor graphs of higher rate codes are subgraphs of lower rate codes [61]. For example, Lin and Yu [62] designed an RC coding scheme for a hybrid ARQ with forward error correction (ARQ/FEC) system, where the transmitter keeps sending additional redundant bits on request until the decoder claims a successful decoding. Having been applied to convolutional codes [61] and turbo codes [63], RC techniques are proven not only to enhance system performance but also to require only low hardware complexity thanks to the structure of a single pair of encoder and decoder.

Since RC-LDPC codes were first considered in [67], there has been a fair amount of work in this area. Ha et al. [68] derived puncturing distributions via asymptotic analysis while assuming infinite block length and no presence of short cycles. Later, in [69], the authors focused on minimizing the number of iterations required to recover punctured bits. Unlike [69], the work reported in [70] tries to maximize the minimum reliability provided via check nodes. Efficiently-encodable irregular LDPC codes along with a puncturing method were derived in [71], where good performance can only be achieved via puncturing degree-2 non-systematic bits. Also, in [72] and [73], protograph-based RC ensembles were implemented for hybrid ARQ applications and systematic construction of punctured LDPC codes was achieved via successive maximization, respectively. Other puncturing methods can be found in [74] to [76], where the authors proved the existence of a puncturing threshold with an improved decoding algorithm, or enhanced the performance at high SNRs by grouping nodes. On the other hand, extension methods, [67] and [77] have been applied by adding extra parity-check bits to increase the size of the parity-check matrix of the mother code. As a result, lower rate codes are generated based on a high rate mother code. Another strategy to construct lower rate RC-LDPC codes has been described in [78] and employs the concept of information shortening.

We are interested in the design of RC-LDPC codes with reduced performance degradation as compared to unpunctured codes with the same rates. In our preliminary work [79], three puncturing schemes are proposed that are able to generate finite-length RC-LDPC codes with good decoding performance at high rates (ranging from 0.5 to 0.9). The first puncturing scheme is a cycle-counting based (CC-based) technique that exploits the algorithm reported in [15] and briefly described in Section 2.3.3 to determine the puncturing pattern. Given a mother code and a target rate, variable nodes having the largest number of girth-length cycles will be punctured first, such that the decoding performance is expected to improve while breaking the shortest cycles. Using a metric for evaluating the EMD, [14] and [19], a second scheme is called approximate cycle EMD based (ACE-based) puncturing which selects the puncturing pattern by considering the cycle length and graph connectivity simultaneously. Additionally, a third scheme relies on a simulation-based greedy search for the best puncturing pattern among many randomly generated patterns. CC-based and ACE-based are shown to be sub-optimal [79] as we observed a performance gap compared to the exhaustive search algorithm. Based on the structure of short cycles and the ACE spectrum, two extension schemes were devised too

in [80]. In this chapter, we expand the descriptions of these techniques with technical analysis as well as a comprehensive set of numerical results in terms of BER, FER and system throughput.

The organization of this chapter is as follows: Section 3.3 explains the system models and basic notation. The proposed puncturing schemes and extension schemes are detailed in Section 3.4 and Section 3.5, respectively. Section 3.6 presents simulation results with explanations. Finally, Section 3.7 concludes this chapter.

3.3 System Model and Basic Notation

This section presents a system model for the proposed puncturing and extension schemes, as well as the design strategy behind them. All of the proposed techniques are based on cycle conditioning for each subgraph (puncturing) or extended graph (extension). Note that traditional cycle conditioning has only focused on eliminating short cycles in a TG with very long block sizes. But it has been proven that avoiding short cycles alone is not enough to achieve good performance, particularly in the error-floor region, [14] and [19], and cycle conditioning is a challenging task for finite-length LDPC codes. Thus, in this work we derive a strategy to generate a family of finite-length LDPC codes over the AWGN channel.

3.3.1 Construction of RC-LDPC Codes Using Puncturing

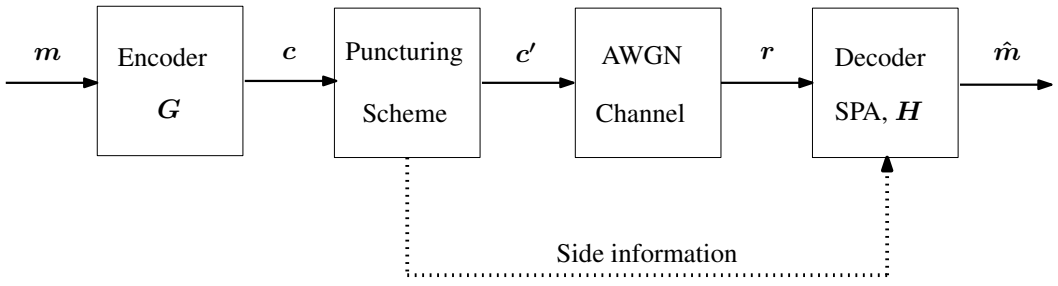

Figure 3.1: System transmission model for puncturing.

Given a mother LDPC code containing K information bits, the code rate is given by

$R = K/N$ where N is the block length. The puncturing model is depicted in Fig. 3.1, where $\boldsymbol{m}$ represents the message from the source, $\boldsymbol{c}$ denotes the encoded data, $\boldsymbol{c'}$ is the punctured data, and $\hat{\boldsymbol{m}}$ denotes the estimate of the original message using a BP decoding algorithm given the received data $\boldsymbol{r}$. Notice that the LLR of a punctured bit is set to 0 at the beginning of the decoding process. Suppose that P bits are punctured before transmission, so the resulting code rate is given by $R' = K/(N - P)$ and the puncturing rate by $\rho = P/N$. We assume that the decoder has perfect knowledge regarding the puncturing pattern, i.e. the position of punctured bits in a codeword. Otherwise, some side information is needed to send the puncturing pattern to the receiver end. Puncturing is a common and simple method to construct RC codes, for which a higher rate is achievable by means of removing a subset of encoded bits $\boldsymbol{c}$ [74]. A randomly chosen puncturing pattern [67] can be used to realize the rate compatibility at the expense of severe performance degradation. Intentional puncturing methods were investigated for short block LDPC codes in [69] - [71] and [76], ranging from asymptotic analysis to grouping and sorting variable nodes. In contrast to those methods, the proposed puncturing schemes aim to diminish the performance loss caused by puncturing from a cycle distribution perspective, i.e. the puncturing pattern is selected in the sense that the removed bits will break a certain number of short cycles, which significantly improves the connectivity of the TG.

3.3.2 Construction of RC-LDPC Codes Using Extension

The authors in [67] and [77] state that extension is another effective approach to constructing good RC-LDPC codes. We also employ the idea of cycle conditioning to devise the proposed extension schemes. The proposed extension framework is built as shown in Fig. 3.2, in which, starting from level 1 and running to level L, the current parity-check matrix is extended in such a way that the same number of rows and columns are added in each level. Consequently, the corresponding code rate gradually reduces. Since $M_l = N_l = B(l = 1, \ldots, L)$, the matrix $\boldsymbol{h}_{\text{ext}}$, along with two accompanying identity matrices, is a $B \times B$ square matrix. Note that $\boldsymbol{h}_{\text{ext}}$ is fixed from one level to the next. In Fig. 3.2, the areas filled by "**0**" ensure the sparseness of the extended parity-check matrix $\boldsymbol{H}_{\text{ext}}$, and the existence of identity matrices guarantees a rather uniform degree of check node distribution as well as createing sufficient dependency between the original matrix

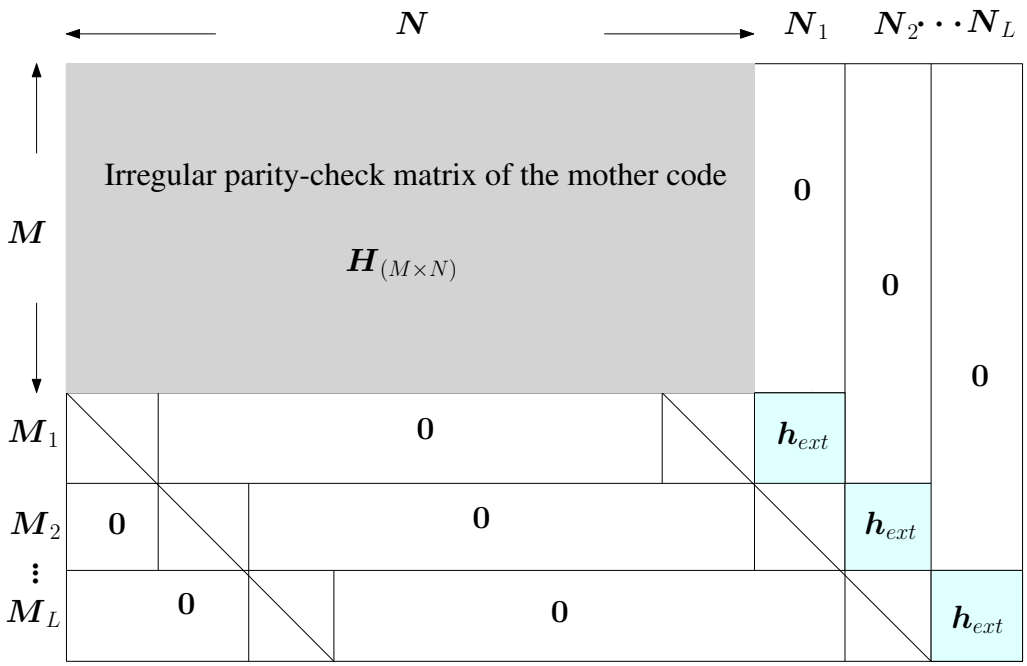

Figure 3.2: Multi-level strategy for the extension of the $M \times N$ parity-check matrix $\boldsymbol{H}$ to the $M_L \times N_L$ matrix $\boldsymbol{H_{ext}}$.

$\boldsymbol{H}$ and the newly-extended matrix $\boldsymbol{H}_{\mathrm{ext}_l}$. Our framework is very similar to that in [77], which enables fast linear-time encoding as matrix $\boldsymbol{H}_{\mathrm{ext}}$ is always obtained in systematic form by using GE. Furthermore, the proposed extension schemes have two extra features: 1) possible cycles of length 4 are avoided by not putting two identity matrices together; 2) more importantly, the submatrices $\boldsymbol{h}_{\mathrm{ext}}$ are carefully chosen with cycle conditioning for each subgraph. The transmission model for extension is illustrated in Fig. 3.3 where $\boldsymbol{I}$ denotes the identity matrix, $\boldsymbol{G}$ the generator matrix of the mother code, and $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_L$ are systematically transformed from the extended counterparts of $\boldsymbol{H}_{\mathrm{ext}1}, \ldots, \boldsymbol{H}_{\mathrm{ext}L}$. Similar to the puncturing model, some side information is needed at the receiver end to indicate the desired rate and corresponding parity-check matrix.

3.4 Proposed Puncturing Techniques

Inspired by the cycle-conditioning and ACE metric, in this section we introduce the proposed puncturing schemes, i.e. CC-based puncturing, ACE-based puncturing, and simulation-based puncturing. The first two methods are developed using the counting cy-

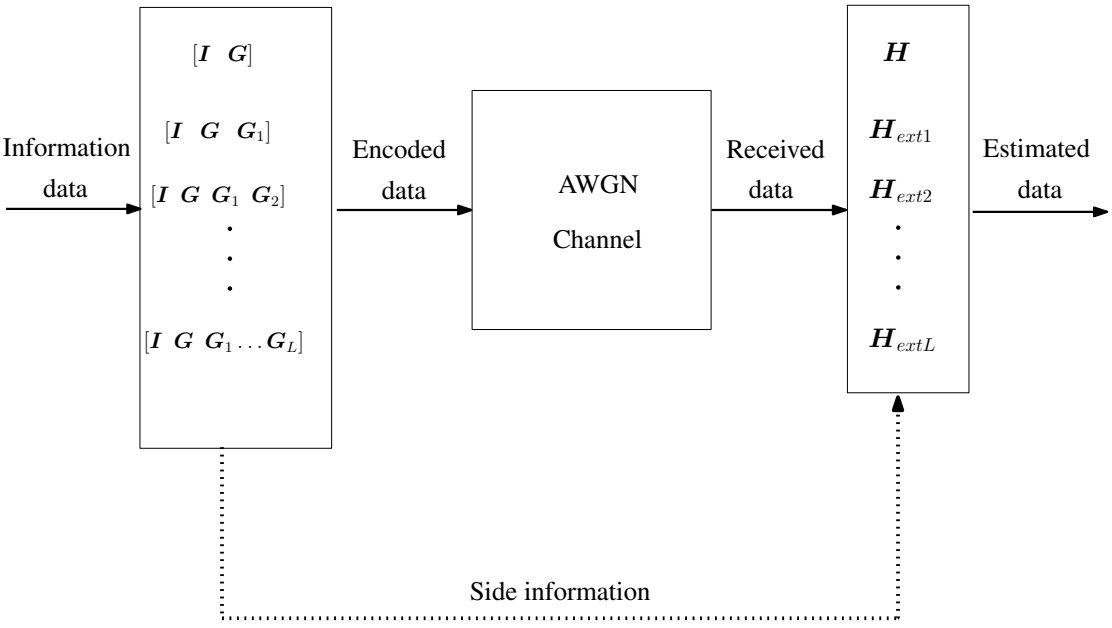

Figure 3.3: System transmission model for proposed extension schemes.

cle algorithm and the ACE metric, while the last one is based on an exhaustive search. As mentioned in the last section, all the proposed puncturing techniques can be applied offline (independent of the data transmitted), and without any side information the puncturing patterns need to be stored at both the transmitter and receiver ends. Unlike the preliminary results reported in [79], we have replaced the cycle counting algorithm [15] with a more complexity-effective algorithm [16], modified the puncturing order of the proposed ACE-based scheme, and employed improved PEG code (ACE PEG) [81] as the mother code.

3.4.1 CC-Based Puncturing Scheme

The proposed CC-based puncturing technique is developed based on the counting cycle algorithms [15] and [16]. The former algorithm employs matrix multiplications while the latter takes advantage of the message passing nature of BP decoding. Given the same TG, we have verified that both algorithms produce similar results for counting cycles of length g and $g+2$, where g is the girth. But the algorithm in [16] has much lower complexity ($\mathcal{O}(g|E|^2)$) than its counterpart [15] ($\mathcal{O}(gN^3)$), especially for graphs with large sizes. Provided with the cycle distribution, the objective is to select an ideal puncturing pattern that can break as many girth-length cycles as possible, which may reduce the performance

degradation caused by puncturing. The idea behind the proposed algorithm is inspired by the fact that the existence of short cycles creates a statistical dependency between the extrinsic messages being exchanged in the current decoding iteration, such that the extrinsic messages for the next iteration will, inaccurately, have high reliability.

According to the PEG algorithm [21], high degree nodes are placed in the leftmost positions of $\boldsymbol{H}_{(M\times N)}$ that correspond to information bits, as they provide more protection for the original data. Following this design rule we only puncture the set of variable nodes $s_j \in V_s$ where $(K+1 \leq j \leq N)$. Define the vector $\boldsymbol{c}_{s_j} = \{N_g, N_{g_2}, N_{g_4}\}^T$ whose element refers to the number of $g-\text{cycles}$, $(g+2)-\text{cycles}$ and $(g+4)-\text{cycles}$ passing through a variable node s_j. Any $s_j(K+1 \leq j \leq N)$ will be included as a puncture candidate if $N_g \neq 0$. For each candidate node, another vector $\boldsymbol{v}_{g\_s_j}$ is formed as:

$$\boldsymbol{v}_{g\_s_j} = \{v_{g\_s_0}, v_{g\_s_1}, \ldots, v_{g\_s_{N-1}}\}^T, \quad s_j \in V_c, \tag{3.1}$$

where entries represent the number of cycles of length g that s_j has, and are arranged in decreasing order. Similarly, we can also define $\boldsymbol{v}_{g_2\_s_j}$ or $\boldsymbol{v}_{g_4\_s_j}$ if necessary. There are two criteria to determine the set of punctured nodes: 1) to find variable nodes having the shortest cycles passing through; 2) to find variable nodes having more such cycles than others. In addition, we also tried to arrange the entries of (3.1) in reverse manner, i.e. start with puncturing the variable nodes having the least number of cycles of length g. But with such a formation, the performance deteriorates dramatically. If candidates on the $g-\text{cycles}$ are less than P, we puncture P nodes at first then arrange the rest of the candidates with respect to the $(g+2)-$ and $(g+4)-\text{cycles}$. But this situation rarely occurs in practice unless an unreasonable puncturing rate ρ is given. Compared to random puncturing schemes, CC-based puncturing requires more computational complexity due to the cycle counting algorithm. On the other hand, CC-based puncturing has been verified as significantly outperforming random puncturing techniques [79]. Obviously, the complexity of CC-based puncturing is mainly increased by counting short cycles. It is worth noting that the practical complexity is lower than $\mathcal{O}(g|E|^2)$ since, most of the time only a counting cycle of length g is required. The proposed CC-based puncturing technique can be summarized thus:

Step 1: given block size N, rate R and the degree distribution, generate the parity-check matrix of mother code $\boldsymbol{H}_{(M\times N)}$ by using improved PEG [81];

Table 3.1: Cycle Distributions of Code A before and after CC-based Puncturing

	Mother code $N = 1000$				Punctured code $N = 800$		
	N_c	μ_c	σ_c		N_c	μ_c	σ_c
$c = 8$	3513	3.5	4	$c = 8$	2037	2.5	14.3
$c = 10$	38553	38.5	37.5	$c = 10$	36482	45.6	138.3
$c = 12$	390337	390	380	$c = 12$	371223	464	1462

Step 2: for $\boldsymbol{H}_{(M\times N)}$ compute $g-$ cycle, $(g+2)-$ cycle and $(g+4)-$ cycle with respect to variable nodes $s_j \in V_s$ where $(K+1 \leq j \leq N)$;

Step 3: based on the knowledge from Step 2, define vector $\boldsymbol{c}_{s_j} = \{N_g, N_{g_2}, N_{g_4}\}^T$ for every variable node s_j. If $N_g \neq 0$, s_j is picked as one of the puncture candidates;

Step 4: for all the candidates chosen in Step 3, define the vector $\boldsymbol{v}_{g\_s_j}$ $(s_j \in V_s)$. Puncture the first P candidates in $\boldsymbol{v}_{g\_s_j}$.

Now we illustrate how CC-based puncturing affects the cycle distribution as well as overall performance. As for a cycle of length c, the cycle distribution is defined as (N_c, μ_c, σ_c) where N_c denotes the number of cycles of length c, while μ_c and σ_c denote the mean and standard deviation of c-length cycles with respect to the variable nodes. By way of example, we use an irregular TG of code A in Section 3.6, which has block length of $N = 1,000$, code rate $R = 0.5$ and degree distributions $\upsilon(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$, $\nu(x) = x^5$. Table 3.1 shows the cycle distributions of the mother code and the punctured code. Applying CC-based puncturing, short cycles of length $g = 8$ are reduced by 1476 while short cycles of length 10 and 12 are reduced by 2071 and 19114, respectively. Even if the number of girth-length cycles diminishes, it is worth noticing that the cycle distribution becomes less uniform after puncturing. In [15], the authors suggest that codes with a uniform cycle distribution perform better than codes of the same girth but with a non-uniform cycle distribution. As a consequence, the proposed CC-based puncturing removes a fair number of cycles of girth length but also damages the inherent connectivity of the TG. Based on this fact, we

are motivated to devise more advanced puncturing scheme as a sequel.

3.4.2 ACE-Based Puncturing Scheme

The second puncturing algorithm proposed is an improved version of the CC-based puncturing scheme, which is called an ACE-based puncturing algorithm, thanks to the employment of an ACE metric. ACE-based puncturing strives to remove a certain type of short cycles and simultaneously maintain good graph connectivity. Since not all short cycles of the same length are equally detrimental to iterative decoding, the ACE metric [14] and ACE spectrum [19] were introduced to evaluate the consequences of short cycles with a certain length in a TG. For a cycle $\mathfrak{C}$ and a corresponding set of variable nodes $V_{\mathfrak{C}}$, all the edges connected to $\mathfrak{C}$ can be categorized into three groups [19]:

- $E_{\text{cyc}}(V_{\mathfrak{C}})$: cycle edges within cycle $\mathfrak{C}$.
- $E_{\text{cut}}(V_{\mathfrak{C}})$: cut edges incident to check nodes not in $\mathfrak{C}$ but at least doubly connected to $V_{\mathfrak{C}}$.
- $E_{\text{ext}}(V_{\mathfrak{C}})$: extrinsic edges incident to those check nodes with a single connection to $V_{\mathfrak{C}}$.

Ideally, the set $E_{\text{ext}}(V_{\mathfrak{C}})$ is expected to be large so that short cycles will possess more singly connected extrinsic edges, which decreases the probability of cycles forming a small stopping [65] or trapping set [82]. According to [14], $|E_{\text{ext}}(V_{\mathfrak{C}})|$ of a cycle $\mathfrak{C}$ can be approximated by using:

$$\epsilon_{ACE} = \Sigma_{s_j \in V_{\mathfrak{C}}} d_{s_j} - 2, \tag{3.2}$$

where d_{s_j} denotes the degree of the variable node s_j and $s_j \in V_{\mathfrak{C}}$. For short cycles of the same length, a larger ACE value indicates better connections to the rest of the graph. Here we define the average ACE value regarding a variable node s_j contained in N_g cycles of length g as:

$$\alpha_g = 1/N_g \sum_{n_c=1}^{N_g} \epsilon_{ACE}^{(n_c)}, \tag{3.3}$$

where α_{g2} and α_{g4} are defined with respect to cycles of length $g+2$ and $g+4$. Moreover, for each $s_j \in V_s$ where $(K+1 \leq j \leq N)$, α_{s_j} is defined as:

$$\alpha_{s_j} = \min\{\alpha_g, \alpha_{g2}, \alpha_{g4}\}. \tag{3.4}$$

Compared to the work reported in [79], the ACE puncturing proposed has the following three improvements: 1) the puncturing ordering is adjusted to consider the connectivity of cycles rather than their length; 2) a new code design [81] for generating the mother code makes indexing ACE values more convenient; 3) the combination of a new design method and ordering leads to improved performance for both mother code and punctured code. The proposed ACE puncturing can be depicted as follows:

Step 1: given block size N, rate R, and the degree distributions, generate the parity-check matrix for the mother code $\boldsymbol{H}_{(M\times N)}$ by using the improved PEG [81];

Step 2: for $\boldsymbol{H}_{(M\times N)}$ compute $g-\text{cycle}$, $(g+2)-\text{cycle}$ and $(g+4)-\text{cycle}$ for the variable nodes $s_j \in V_s$ where $(K+1 \leq j \leq N)$;

Step 3: with the knowledge from Step 2, define the vector $\boldsymbol{c}_{s_j} = \{N_g, N_{g_2}, N_{g_4}\}^T$ for every variable node s_j ($s_j \in V_s$). Calculate α_{s_j} using (3.3) and (3.4);

Step 4: find the set of puncturing candidates $\boldsymbol{w} = \{\alpha_{s_0}, \alpha_{s_1}, \ldots, \alpha_{s_{N-1}}\}^T$ by sorting α_{s_j} in increasing order;

Step 5: puncture the first P candidates in $\boldsymbol{w}$.

We use Table 3.2 to illustrate the change in the cycle distribution after running the ACE puncturing scheme. Compared to the results from Table. 3.1, ACE puncturing is able to maintain a relatively uniform cycle distribution by first removing the variable nodes which get involved with longer cycles but have low ACE values. From the decoding point of view, in a subgraph with good connectivity, the LLR of punctured bits is expected to

Table 3.2: Cycle Distributions of Code A before and after ACE-based Puncturing

	Mother code $N = 1000$				Punctured code $N = 800$		
	N_c	μ_c	σ_c		N_c	μ_c	σ_c
$c = 8$	3513	3.5	4	$c = 8$	2968	3.7	4.8
$c = 10$	38553	38.5	37.5	$c = 10$	31577	39.4	59.5
$c = 12$	390337	390	380	$c = 12$	323560	404	609

be recovered within a few iterations, even though there might be other punctured bits in the same neighborhood. On the other hand, unlike CC puncturing, ACE puncturing does not work for regular codes since all the ACE values of variable nodes are identical. In that case, it is impossible to consider puncturing priority with the ACE metric.

3.4.3 Simulation-Based Puncturing Scheme

The last puncturing scheme proposed is developed on the basis of an exhaustive search among a large number of random puncturing patterns. Then, the best puncturing pattern is determined simply by choosing the one having the best average BER performance. At the receiver end, in order to find the best pattern, we need to send r training sequences then compute the average BER values at T SNR points for each puncturing pattern. The reason we sample BER at various SNR points is because the performance of LDPC codes may vary in different SNR regimes [7], so that the optimised performance is obtained on average over these SNR regimes. For Q possible patterns the best pattern $\boldsymbol{p}_{\text{opt}}$ is selected as:

$$\boldsymbol{p}_{\text{opt}} = \arg\min_{q} \frac{1}{rT} \sum_{i=1}^{r} \sum_{t=1}^{T} \text{BER}(\boldsymbol{p}_q), \quad q = 1, \ldots, Q. \tag{3.5}$$

The proposed simulation-based (SIM-based) algorithm can be described as follows:

Step 1: given block size N, rate R and the degree distributions, generate the parity-check matrix of the mother code $\boldsymbol{H}_{(M\times N)}$ by using the improved PEG [81];

Step 2: for the desired rate R', randomly generate Q puncturing patterns represented by a row vector $\boldsymbol{p}_q$ where $q = 1, \ldots, Q$, in each of which P bits are randomly punctured from the encoded data;

Step 3: for each pattern in $\boldsymbol{p}_q$, send a training sequence of length $1,000$ at T SNR points then calculate BER values;

Step 4: after running r repetitions, for all Q patterns calculate an average based on accumulated BER values;

Step 5: select the best puncturing pattern $\boldsymbol{p}_{\text{opt}}$ among $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_Q$ by choosing the pattern with the minimum average BER.

From (3.5), it is apparent that given a desired rate R' it is possible to obtain the optimal pattern p_{opt} when all $\frac{N!}{M!(N-M)!}$ possible puncturing patterns are considered, which seems infeasible in practice. Since the quality of the best pattern $\boldsymbol{p}_{\text{opt}}$ depends on Q, the last proposed puncturing scheme offers flexible trade-offs between performance and the number of candidate patterns. In [79], SIM-based puncturing always outperforms CC-based puncturing and ACE-based puncturing. Nevertheless, with the additional improvement, ACE-based puncturing is able to provide at least comparable performance to SIM-based puncturing, even when we increase Q to 500.

3.5 Proposed Extension Techniques

In this section, we investigate another pathway to generate RC LDPC codes, i.e. extension techniques, and two proposed schemes are explained in the sequel. An extension framework introduced in [77] is exploited which enables fast encoding and off-line operation for the proposed extension schemes. To refine the techniques described in [80], we replace the cycle counting algorithm [15] by a more efficient algorithm [16], further

develop the design process, and utilize the improved PEG code (ACE PEG) [81] as the mother code.

3.5.1 Counting-cycle based extension

The first extension scheme proposed is the CC-based extension which employs an algorithm for counting short cycles in order to select extension submatrix $\boldsymbol{h_{ext}}$ among S candidates. We set the parameter S to equal the number of desired extension rates, e.g. $S = 3$ if $R = 5/10$, $R_1 = 5/12$ and $R_2 = 5/13$, where R is the rate of the mother code, although S can be set to a larger value that possibly allows us to find a better extension matrix $\boldsymbol{h}_{ext}$, which will be defined below. In this case, three distinct submatrix candidates are constructed using the ACE PEG algorithm [81] with different degree distributions, which are derived via DE [28] and provided with the maximum variable nodes' degree d_v and check nodes' degree d_c. As per the extended part in Fig. 3.2, the $B \times B$ submatrix $\boldsymbol{h}_{\text{ext}}$ is very likely to have many short cycles while the rest of extended part can be proved cycle free. Define $g_{h(s)}(s = 1, 2, \ldots, S)$ as the local girths for each candidate submatrix, and $N_{g(s)}$ as the number of cycles of length $g_{h(s)}$ corresponding to each $\boldsymbol{h}_s(s = 1, 2, \ldots, S)$. After running the counting cycle algorithm [16], we select the candidate submatrix with the largest g_h and the smallest N_g as $\boldsymbol{h}_{ext}$. As a result, the CC-based extension scheme maximizes the local girth g_h of $\boldsymbol{h}_{ext}$, and the selected $\boldsymbol{h}_{ext}$ has the smallest number of length-g_h cycles. The algorithm flow of the proposed CC-based extension is summarized as follows:

Step 1: provided with the parity-check matrix $\boldsymbol{H}_{(M\times N)}$ and the desired code rates $R_1, R_2, ..., R_L$, determine the number of extension levels L which ensures $M_l = N_l = B(l = 1, \ldots, L)$;

Step 2: set $S = L + 1$, given d_v and d_c derive S degree distributions according to DE;

Step 3: based on Step 2, construct S candidates for $B \times B$ submatrices by using the improved PEG algorithm [81];

Step 4: for each submatrix candidate compute the $g_{h(s)}$ and $N_{g(s)}$ of each subgraph;

Step 5: choose the candidate with the largest $g_{h(s)}$ and the smallest $N_{g(s)}$ as $\boldsymbol{h}_{\text{ext}}$;

Step 6: for $1 \leq l \leq l$ gradually extend $\boldsymbol{H}_{(M\times N)}$ to $\boldsymbol{H}_{\text{ext}(L)}$ by adding zero entries, identity matrices, and $\boldsymbol{h}_{\text{ext}}$ as in Fig. 3.2.

3.5.2 ACE-based extension

The second scheme proposed is an ACE-based extension scheme. Unlike CC-based extension, the candidate submatrix with the largest $\alpha(g_h)$ will be selected as the $\boldsymbol{h}_{\text{ext}}$, where $\alpha(g_{h(s)})$ is the average ACE spectrum with respect to $N_{g(s)}(s = 1, 2, \ldots, S)$, as defined in (3.3). Similar to ACE-based puncturing, it is straightforward to compute $\alpha(g_h)$ if the submatrix candidates are created using the ACE PEG algorithm, [81]. The proposed ACE-based extension is described in the following:

Step 1: provided with the parity-check matrix $\boldsymbol{H}_{(M\times N)}$ and the desired code rates $R_1, R_2, ..., R_L$, determine the number of extension levels L which ensures $M_l = N_l = B(l = 1, \ldots, L)$;

Step 2: set $S = L + 1$, given d_v and d_c derive S degree distributions according to DE;

Step 3: based on Step 2, construct S candidates for $B \times B$ submatrices by using the improved PEG algorithm [81];

Step 4: for each submatrix candidate compute $g_{h(s)}$ and $N_{g(s)}$ for each subgraph;

Step 5: provided with $g_{h(s)}$ and $N_{g(s)}$ in Step 3, calculate $\alpha(g_{h(s)})$;

Step 5: choose the candidate with the largest $\alpha(g_{h(s)})$ as $\boldsymbol{h}_{\text{ext}}$;

Step 6: for $1 \leq l \leq l$ gradually extend $\boldsymbol{H}_{(M\times N)}$ to $\boldsymbol{H}_{\text{ext}(L)}$ by adding zero entries, identity matrices, and $\boldsymbol{h}_{\text{ext}}$ as in Fig. 3.2.

In the next section, it will be shown that ACE-based extension offers better performance than CC-based extension. Nevertheless, similar to the circumstances for puncturing, CC-based extension can be used for both regular and irregular codes, while ACE-based extension works specifically for irregular LDPC codes.

which shows that the former performs better at higher rates while the latter is superior at lower rates.

3.6 Simulation Results

First, this section presents numerical results corresponding to the three forms of proposed puncturing and two proposed extension algorithms, respectively. Then, joint comparisons are carried out of the puncturing and extension schemes at different rates. In all the simulations, mother codes are finite-length irregular LDPC codes generated by the improved PEG algorithm [81]. Code A has block length of $N = 1,000$, code rate $R = 0.5$ and degree distributions $\upsilon(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$, $\nu(x) = x^5$. Code B has blocklength $N = 2,000$, code rate $R = 0.4$ and degree distributions $\upsilon(x) = 0.45 \times x^9 + 0.26 \times x^2 + 0.29 \times x$, $\nu(x) = x^5$. The decoder applies the standard BP algorithm in the logarithm domain.

To test the proposed puncturing schemes, we first choose code A as the mother code,

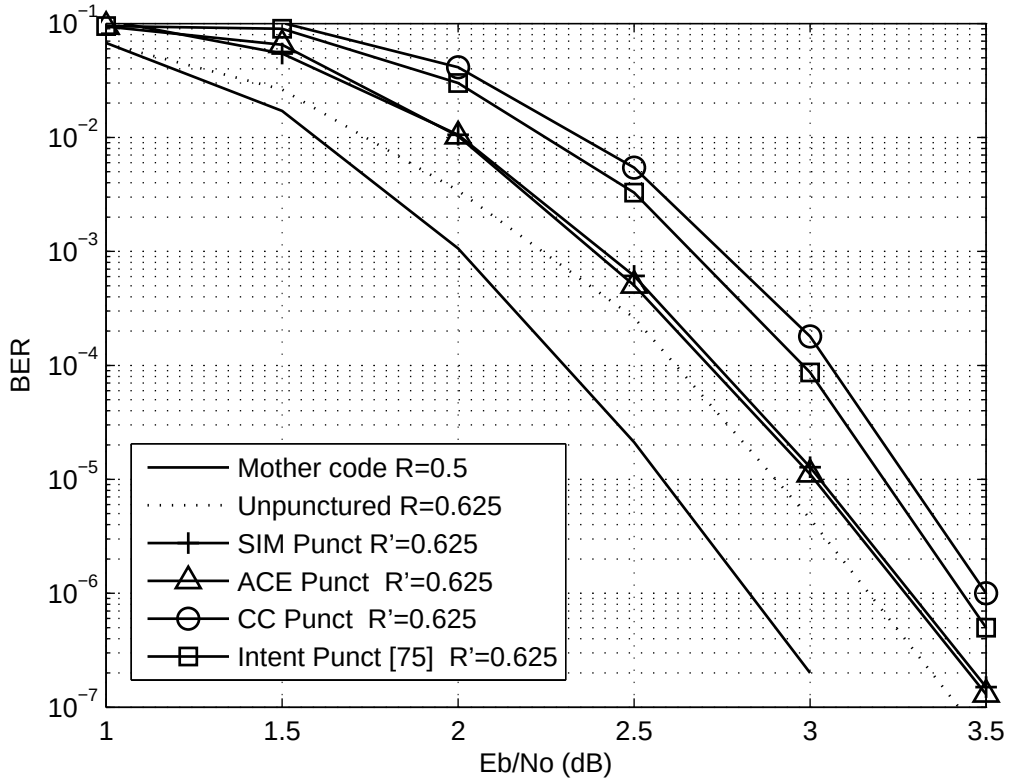

Figure 3.4: Comparisons of the proposed puncturing schemes with an existing puncturing scheme [76] with respect to BER performance, where code A is the mother code.

then compare the performance to that of the puncturing technique reported in [76]. In this scenario, the decoder runs a maximum of 60 decoding iterations. Fig. 5.6 shows a BER performance comparison of the three proposed schemes with the existing method [76], in which the resulting rate R' is 0.625 and the puncturing rate $\rho = 0.2$, such that 200 bits are punctured prior to transmission. It is clear to see that the proposed ACE-based puncturing significantly outperforms the existing method as well as CC-based puncturing and slightly surpasses SIM-based puncturing. For comparison purposes, we also include unpunctured irregular LDPC code, with $N = 800$, $R = 0.625$, which has the same degree distributions as the mother code, A. Notice that the performance gap between ACE-based puncturing and the unpunctured code is less than 0.2 dB at BER of 10^{-6}. Additionally, an FER performance comparison is presented in Fig. 3.5, where only ACE puncturing is included to avoid overlapping curves from other schemes. At each SNR point, we accumulate at least 100 frame errors. It is obvious that ACE-based puncturing renders better FER

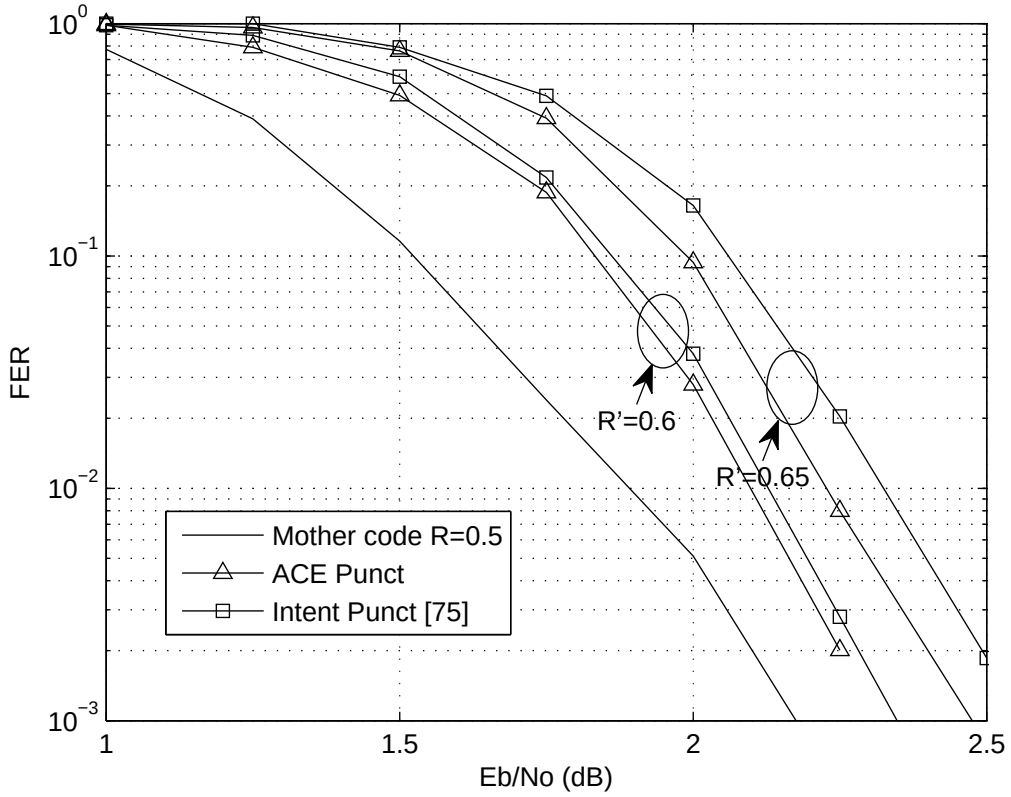

Figure 3.5: Comparison of the proposed ACE puncturing scheme with an existing puncturing scheme in [76] with respect to FER performance, where code A is the mother code.

performance for both resulting rates of $R' = 0.6$ and $R' = 0.65$. More comprehensive results over a range of puncturing rates are shown in Figs. 3.6, 3.7, and 3.8. In Fig. 3.6, CC-based puncturing begins to outperform [76] by 0.25 dB at BER 10^{-5} after the resulting rate of 0.625. Since the performance of [76] dramatically degrades beyond the puncturing threshold $R' = 0.65$, an additional algorithm needs to be applied to achieve good performance at higher rates.

More comparisons with an existing technique are shown in Figs. 3.9 and 3.10, in which we compare the proposed ACE-based puncturing with the puncturing scheme in [71] in terms of BER and FER performance, respectively. In this case, code B is used as the mother code and the maximum number of decoding iterations is increased to 200. From Figs. 3.9 and 3.10, we see that the puncturing scheme of [71] works better in the low SNR region but is outperformed by ACE-based puncturing in the high SNR region. The

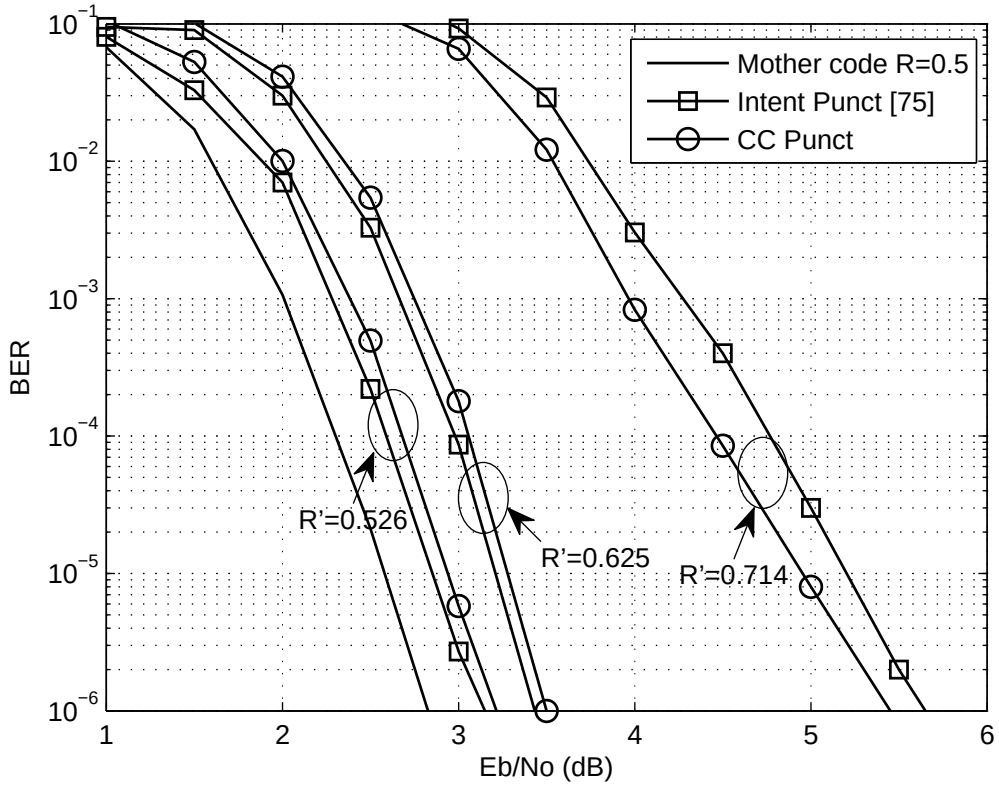

Figure 3.6: Comparison of the proposed CC-based puncturing with an existing puncturing scheme [76] at different resulting rates. R is the rate of the mother code, A, and R' is the resulting rate.

advantages of ACE-based puncturing over [71] are as follows:

- the method [71] is easy to implement for hardware, thanks to a specific code structure. But it usually has to compromise on the optimal degree distribution so as to fulfil the design requirement that may affect the performance. ACE-based puncturing is a more general technique, and can be applied to any irregular mother code;

- the ACE-based method aims to cover a good ACE spectrum via puncturing so graph connectivity is always taken into account at each rate. Due to the design nature of [71], one has to maximize the number of degree 2 variable nodes whose ACE value is 0. Once a cycle is formed, that will severely reduce performance,

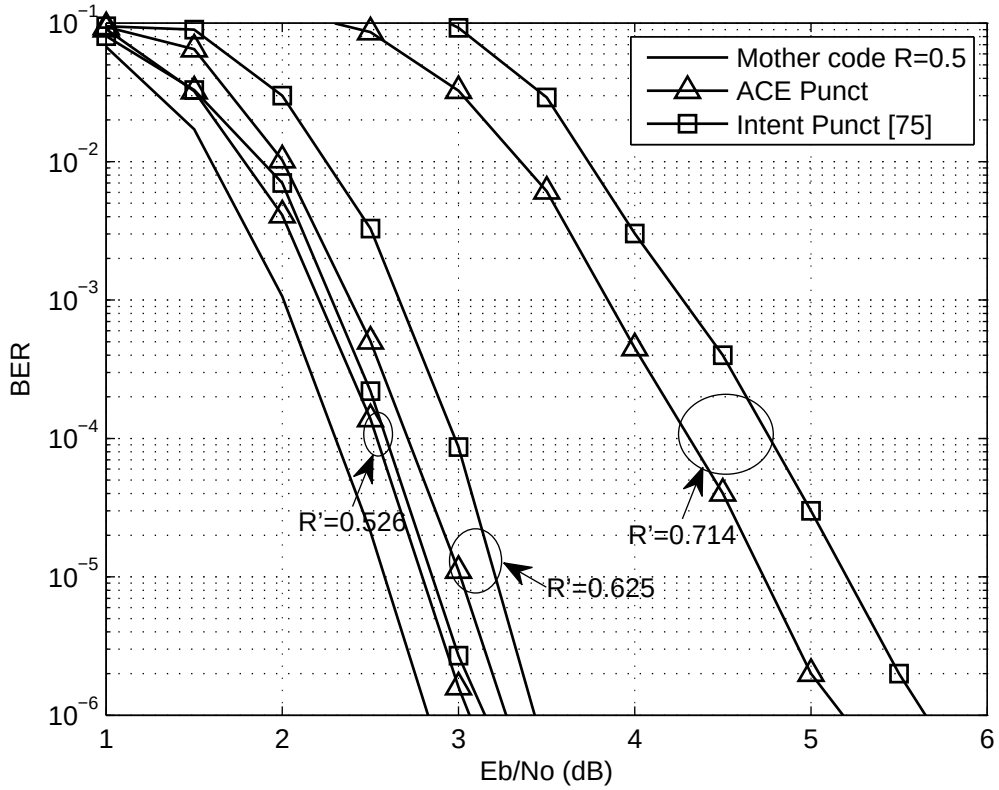

Figure 3.7: Comparisons of the proposed ACE-based puncturing with an existing puncturing scheme [76] at different resulting rates. R is the rate of the mother code, A, and R' is the resulting rate.

especially in the high SNR region;

- the best puncturing performance for [71] results from $N_v(2) = M - 1$ where $N_v(2)$ is the number of degree 2 nodes. However, this requirement is difficult to realize for a mother code with a low rate;

- the ACE scheme is expected to achieve any puncturing rate without limitations, while method [71] always has a puncturing threshold of $R_H = K/(N - N_v(2))$, above which one can only use random puncturing to achieve a higher rate.

As for the extension schemes, we compare the proposed extension techniques with

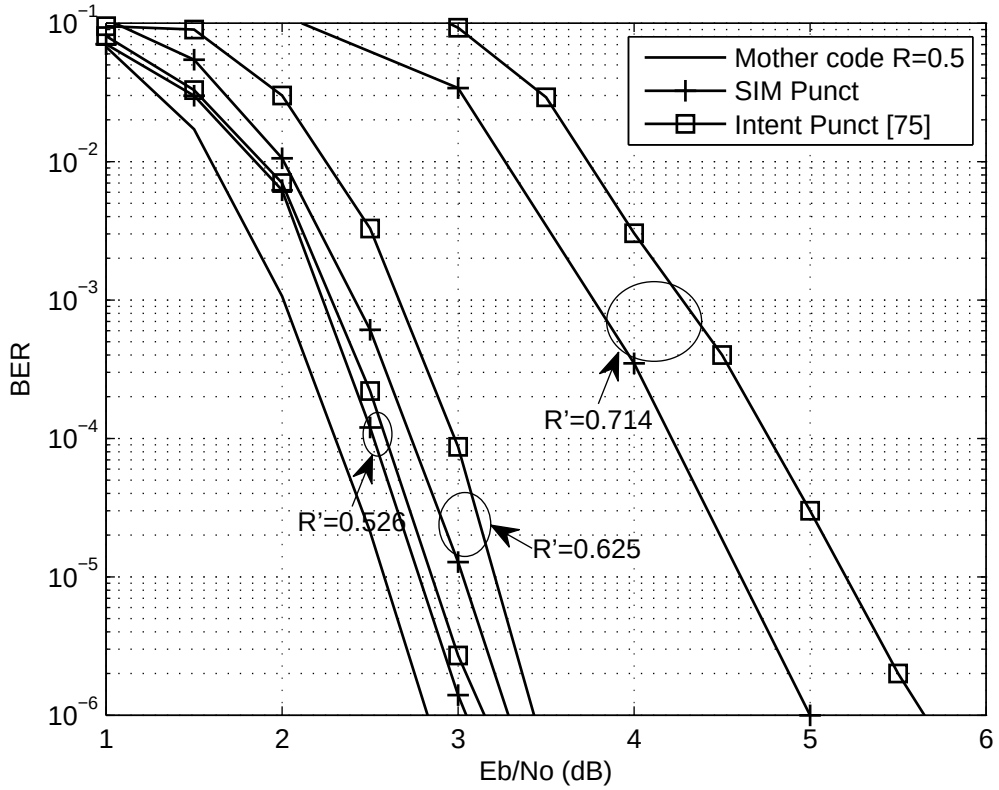

Figure 3.8: Comparison of the proposed SIM-based puncturing with an existing puncturing scheme [76] at different resulting rates. R is the rate of the mother code, A, and R' is the resulting rate.

the technique reported in [77]. In the following simulations, we use the mother code, C, constructed by improved PEG [81] with block length $N = 1,000$, $R = 5/10$ and degree distributions $\upsilon(x) = 0.438x^6 + 0.416x^2 + 0.315x$ and $\nu(x) = 0.561x^6 + 0.438x^5$. The decoder terminates after a maximum of 100 iterations.

In Fig. 3.11, we compare the proposed CC-based and ACE-based extension algorithms with the existing extension method [77]. From rate $5/10$ to $5/14$, the extension operates at three levels and 100 bits are added per level. Notice that all the degree distributions are constrained by $d_{v_{\max}} \leq 7$. In Fig. 3.11, both proposed schemes outperform the existing method at different rates, and the performance gap increases as more parity bits are inserted. On the other hand, the proposed extension schemes are tested for FER performance over a series of blocklengths in Fig. 3.12, which shows that both proposed

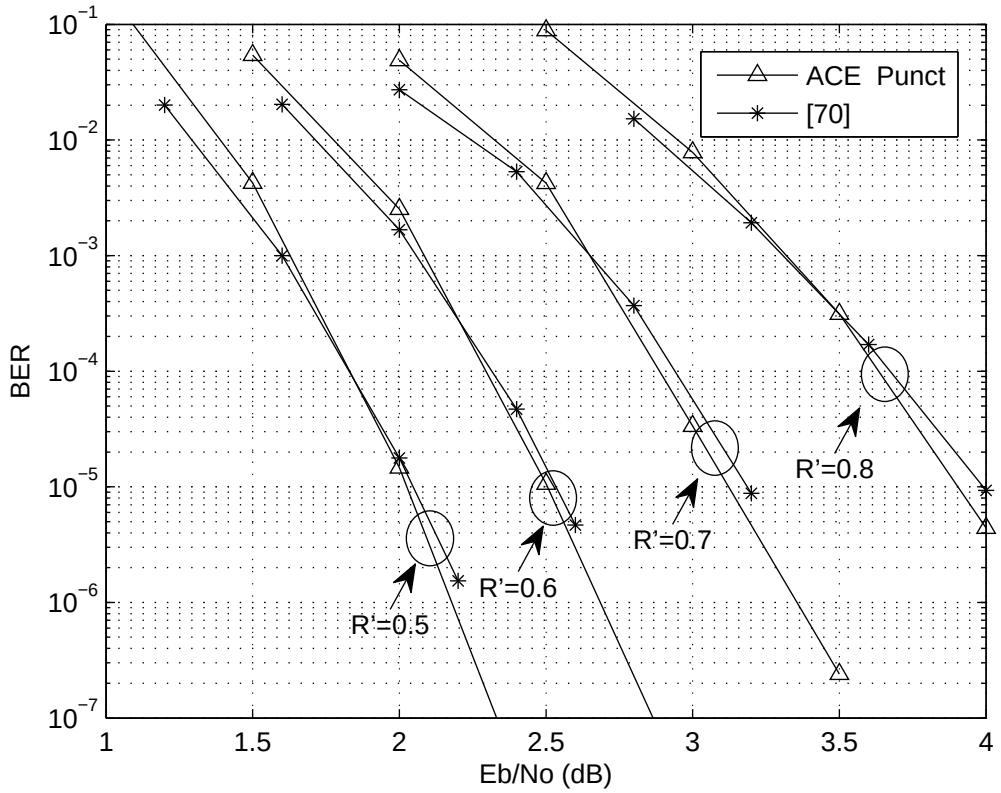

Figure 3.9: A comparison of the puncturing BER performance between ACE-based puncturing and [71]. The puncturing rates R' are 0.5, 0.6, 0.7 and 0.8. Here we use code B as the mother code.

extension schemes manage to perform well with distinct short/moderate blocklengths. For comparison purposes, original codes (without extension) are included with the same block length and rate in Figs. 3.11 and 3.12.

Fig. 3.13 shows that the proposed extension schemes outperform the proposed puncturing schemes at low rates. Both CC-based and ACE-based extensions originates from the mother code, C, while all punctured schemes are from the mother code $(M = 500, N = 1400, R = 5/14)$. On the other hand, Fig. 3.14 shows that at a high rate, $5/8$ of the punctured codes (mother code C) offer better performance as compared to the proposed extension codes whose mother code has $M = 500, N = 700, R = 5/7$. To illustrate the overall performance of the proposed RC-LDPC codes, we finally compare the proposed RC-LDPC codes with the existing RC-LDPC family in the system through-

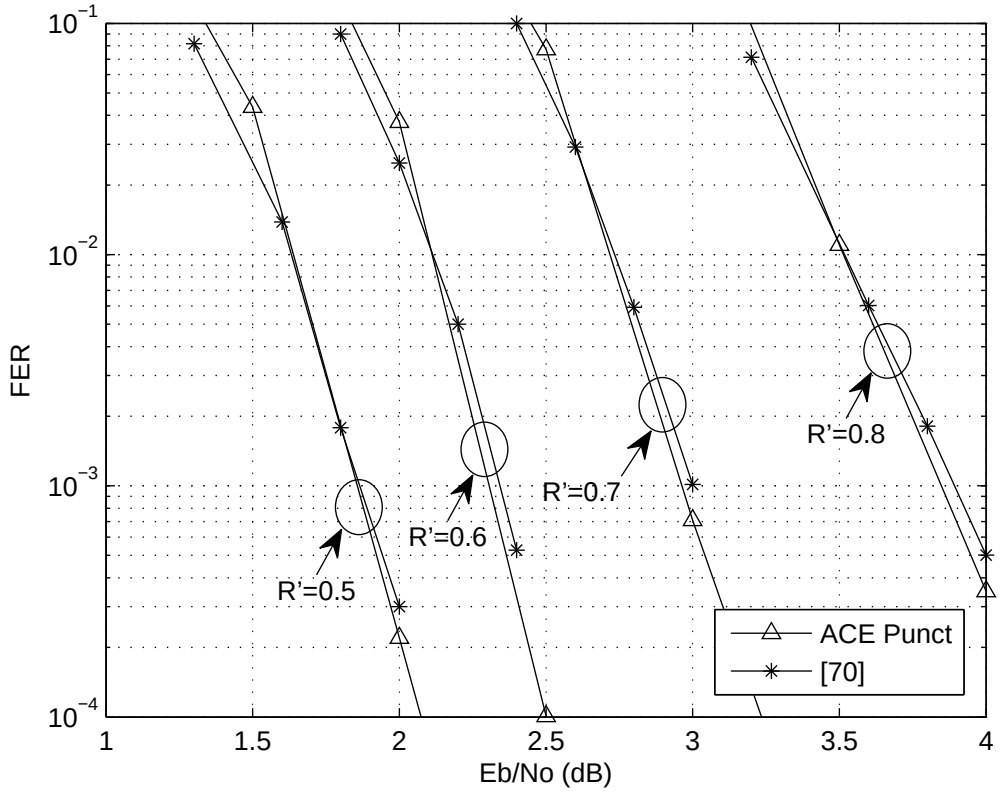

Figure 3.10: The comparison of the puncturing FER performance between ACE-based puncturing and [71]. The puncturing rates R' are 0.5, 0.6, 0.7 and 0.8. We use the mother code B with block length $N = 2,000$ and rate $R = 0.4$.

put [83] as shown in Fig. 3.15, in which E_b in previous figures is replaced by E_s, the average energy per transmitted symbol. Fig. 3.15 shows that the proposed RC-LDPC codes are superior to existing RC codes [77] and can approach channel capacity.

3.7 Summary

In this chapter, we have investigated irregular RC-LDPC codes from both puncturing and extension perspectives. By applying counting cycle algorithms, the ACE spectrum and exhaustive searches, three puncturing schemes as well as two extension schemes have been devised. All proposed schemes manage to achieve various resulting rates, and at

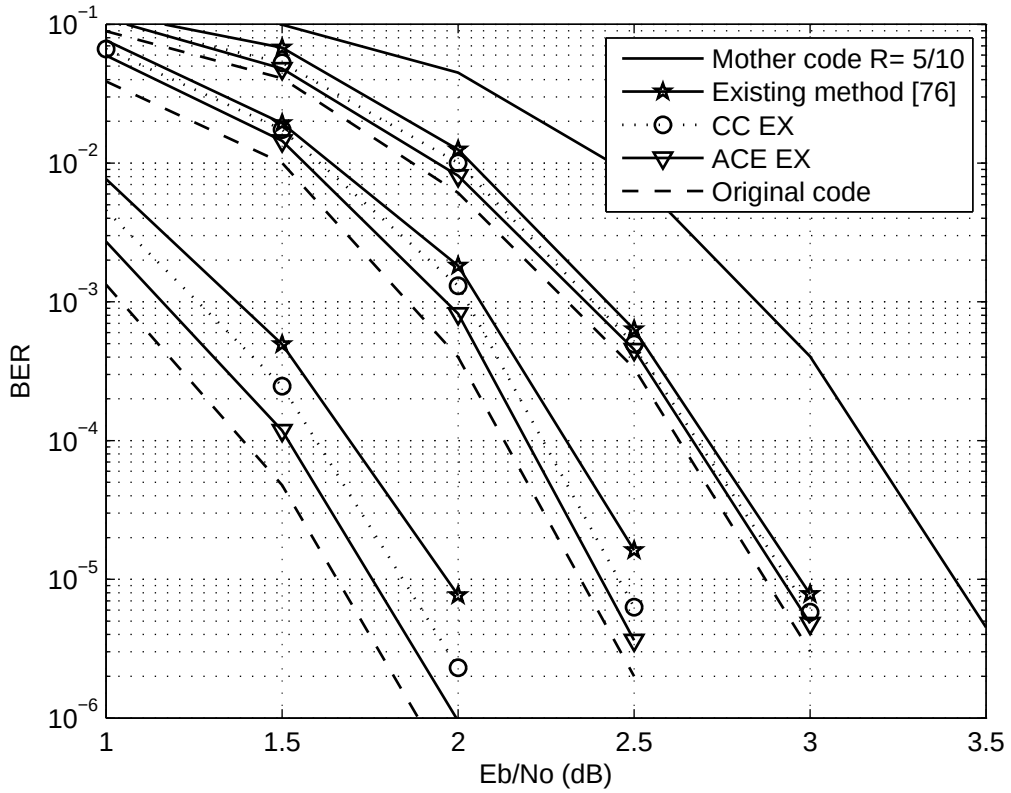

Figure 3.11: Comparison of the proposed extension schemes with another existing scheme [77] at different rates for irregular PEG LDPC codes. The mother code corresponds to the rightmost curve with $N_0 = 1,000$ and $R = 5/10$. For other codes the rates from left to right are $5/14$, $5/13$, $5/12$.

the same time provide better performance than existing methods. Simulation results have shown that the puncturing schemes performs better at higher rates while the extension schemes are superior at lower rates. Given a mother code with rate $R = 0.5$, the proposed extension designs are suitable for creating RC-LDPC with low rates $(R < 0.5)$ and ACE-based extension performs better than the CC-based extension. On the other hand, the puncturing designs are preferred for codes with high rates. With the additional improvement, the ACE puncturing has been proven to generate the optimal puncturing pattern and slightly outperform simulation-based puncturing. As a consequence, taking advantage of a combined puncturing/extension strategy, we have devised algorithms to generate RC-LDPC codes with a wide range of rates $(0.1 < R < 0.9)$. From the rich literature of LDPC codes, it has been widely recognised that exploring the graphical structure can

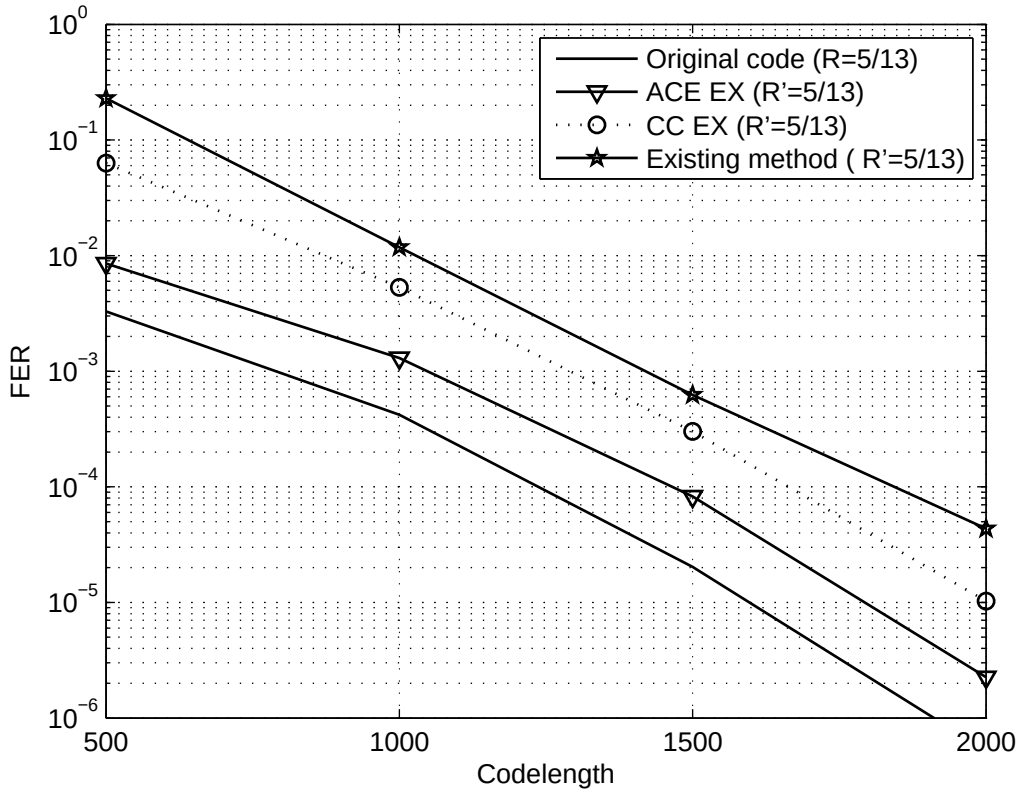

Figure 3.12: Comparisons of the proposed extension schemes with another existing scheme [77] for different block lengths at rate $5/13$ and SNR= 2 dB for irregular LDPC codes.

improve the performance not only at the encoding end but also at the decoding end. In the next chapter, we will move our attention to investigation of the impact of short cycles for decoders.

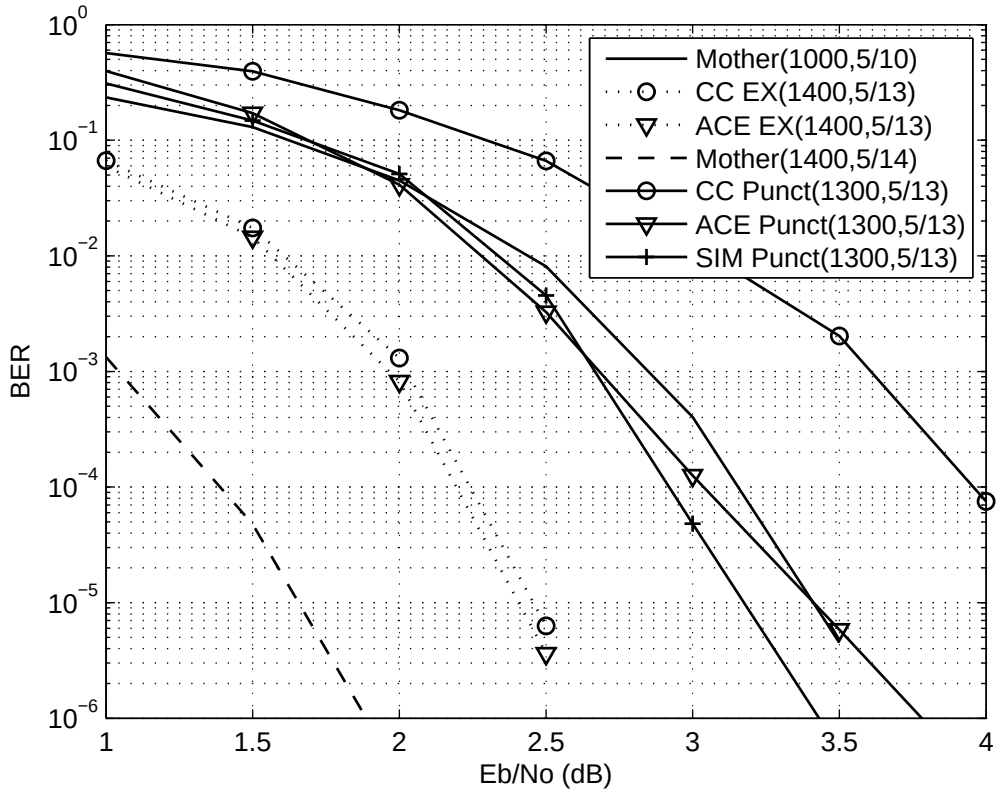

Figure 3.13: Comparison of the proposed extension schemes with the proposed puncturing schemes at a low rate, $5/13$, for irregular LDPC code.

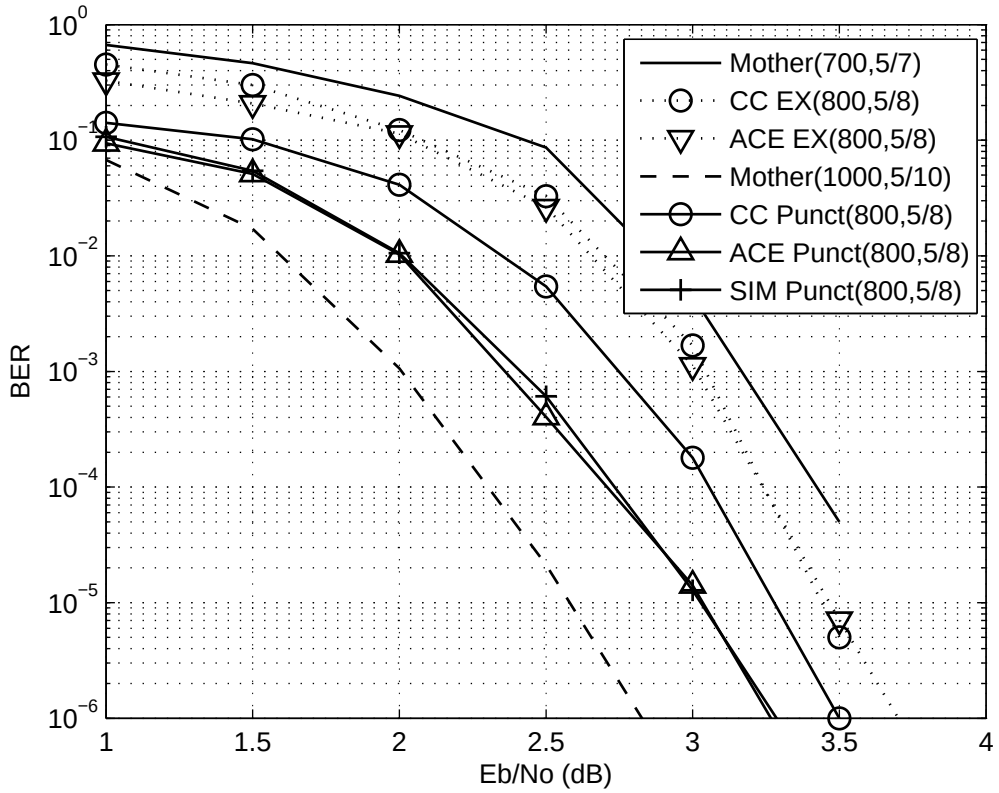

Figure 3.14: Comparison of the proposed extension schemes with the proposed puncturing schemes at a high rate, $5/8$, for irregular LDPC code.

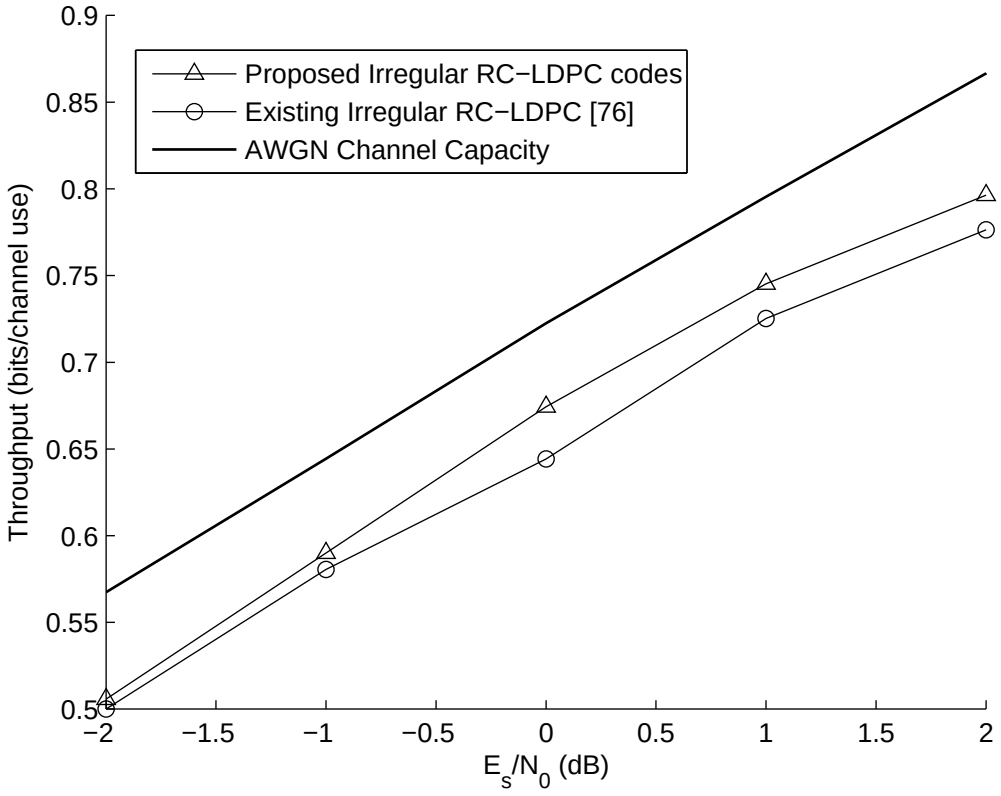

Figure 3.15: Comparison of the proposed irregular RC-LDPC codes with the irregular RC-LDPC code [77] in system throughput. The capacity of AWGN channel is also included.

Chapter 4

Low-Latency Variable Factor Appearance Probability Belief Propagation Decoding for LDPC Codes

Contents

4.1 Overview

In this chapter we propose a novel message passing algorithm which exploits the existence of short cycles to obtain performance gains by reweighting the factor graph. The proposed decoding algorithm is called a VFAP-BP algorithm and is suitable for wireless communications applications with low latency and short blocks. Simulation results show that the VFAP-BP algorithm outperforms the standard BP algorithm, and requires

a significantly smaller number of iterations when decoding either general or commercial LDPC codes.

4.2 Introduction

As mentioned in Chapter 1, LDPC codes are recognized as a class of linear block codes which can achieve near-Shannon capacity with linear-time encoding and parallelizable decoding algorithms. Equipped with efficient decoders, LDPC codes have found applications in a number of communication standards, such as DVB-S2 and Wi-Fi 802.11. Nevertheless, the decoding algorithms of LDPC codes normally require a significantly higher number of iterations than turbo codes, which results in severe decoding latency [9].

The BP algorithm, sometimes also called sum-product algorithm (SPA), is a powerful algorithm to solve inference problems approximately in statistical physics, computer vision, distributed hypothesis testing, cooperative localization and error control coding [7]. This message passing algorithm computes accurate marginal distributions of variables corresponding to each node of a graphical model, and is exceptionally useful when optimal inference decoding is computationally prohibitive due to the large size of a graph. Additionally, the BP algorithm is capable of producing exact inference solutions if the graphical model is acyclic (i.e. a tree), while it does not guarantee convergence if the graph has short cycles which significantly reduce overall performance [17]. Since BP algorithms started to be applied as a decoding algorithm for turbo and LDPC codes, various versions of BP graph-based decoding algorithms have been reported in the area. All relevant decoding strategies, either mitigating the error floor or improving the waterfall behaviour, can be classified into two categories: 1) removing the short cycles in the code graph to avoid "near-codeword" or "trapping sets" that refers to short cycles consisting of nodes with low degrees; 2) enhancing the suboptimal BP decoding algorithm, when using ML decoding is intractable [55]. However, the lack of a convergence guarantee and the high latency due to many decoding iterations are still open issues for researchers when it comes to decoding LDPC codes effectively in wireless communications applications, where a large amount of data transmission and data storage are required. Recently, Wymeersch et al. [84], [20] introduced the uniformly reweighted (URW)-BP algorithm,

which exploits BP's distributed nature and reduces the factor appearance probabilities (FAP) in [17] to a constant value. In [20], the URW-BP has been shown to outperform the standard BP in terms of decoding latency among other applications.

In this chapter, we investigate the idea of reweighting a suitable part of the factorized graph while also, statistically, taking the effect of short cycles into account. By combining the reweighting strategy with the knowledge of the short cycles obtained by the cycle counting algorithms [15], [16], we present the variable FAP (VFAP)-BP algorithm. This algorithm assigns distinct FAP values to each parity-check node on the basis of the structure of short cycles, rather than complex global graphical optimization. We also extend the application of reweighted message passing decoding algorithms from symmetric graphs to asymmetric graphs. Simulation results show that the proposed VFAP-BP algorithm consistently outperforms URW-BP for irregular LDPC codes, and offers better BER performance than the standard BP for both regular and irregular codes when using a small number of iterations. As a result, VFAP-BP considerably improves the convergence behavior of the BP decoder, which allows a lower decoding latency.

The organization of this chapter is as follows: Section 4.3 introduces the system model for the proposed decoder and an overview of a reweighted decoding strategy. In Section 4.4, the proposed VFAP-BP algorithm is presented in detail. Section 4.5 shows the simulation results along with a discussion. Finally, Section 4.6 concludes the chapter.

4.3 System Model and BP's Reweighted Variation

This section introduces the system model used to develop the proposed VFAP-BP algorithm, then, for comparison purposes, briefly explains the URW-BP algorithm, as detailed in [20] and [84].

4.3.1 Graphical Representation of Decoding LDPC Codes

Suppose we have K information bits being transmitted and a set of codewords $\boldsymbol{x}$ with block length N formed by an LDPC encoder, such that the code rate R is K/N and the corresponding parity-check is an $M \times N(M = N - K)$ sparse matrix $\boldsymbol{H}$ containing at least 99% of 0 entries. After transmission, the objective of the decoder is to find an $1 \times N$ estimated codeword $\hat{\boldsymbol{x}}$ which satisfies the parity-check condition $\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \mathbf{0}$. Thus, we can interpret the decoding process as finding $\hat{\boldsymbol{x}} = \arg\max p(\boldsymbol{x}|\boldsymbol{y})$. Using Bayes' rule the a posteriori distribution becomes

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{4.1}$$

where the likelihood ratios $p(\boldsymbol{y}|\boldsymbol{x})$ can be obtained from the channel and $p(\boldsymbol{x})$ is prior information. Nevertheless, directly calculating $p(\boldsymbol{x}|\boldsymbol{y})$ or $p(\boldsymbol{y})$ is computationally prohibitive because of the size of $\boldsymbol{x}$ [17]. For this reason, we resort to BP as a near-optimal message passing algorithm which can compute either $p(\boldsymbol{x}|\boldsymbol{y})$ or $p(\boldsymbol{y})$ approximately.

In the application of decoding, the BP algorithm performs distributed local computations to find an approximate maximum likelihood solution of $p(x_j|\boldsymbol{y})$ for $(j = 0, 1, \ldots, N-1)$. As shown in Fig. 4.2, the LDPC codes can be represented by a factor graph where M square nodes stand for M parity-check equations and N circle nodes relate to N encoded binary bits. There is an edge connecting the check node C_i and the variable node V_j in the factor graph if entry h_{ij} of the parity-check matrix $\boldsymbol{H}$ equals 1. All the check nodes and variable nodes work cooperatively and iteratively to estimate $p(x_j|\boldsymbol{y})$ for $(j = 0, 1, \ldots, N-1)$ [12]. Following a set of message passing rules, the variable nodes (check nodes) process the incoming message and send the extrinsic information to their neighboring check nodes (variable nodes), back and forth in an iterative fashion, until all M parity check conditions are met ($\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \mathbf{0}$) or the decoder reaches the maximum number of iterations.

4.3.2 URW-BP Algorithm for High-Order Interactions

When the factor graph is a tree with no cycles, the standard BP algorithm is able to perform an accurate approximation in a few iterations. In the presence of cycles, it nor-

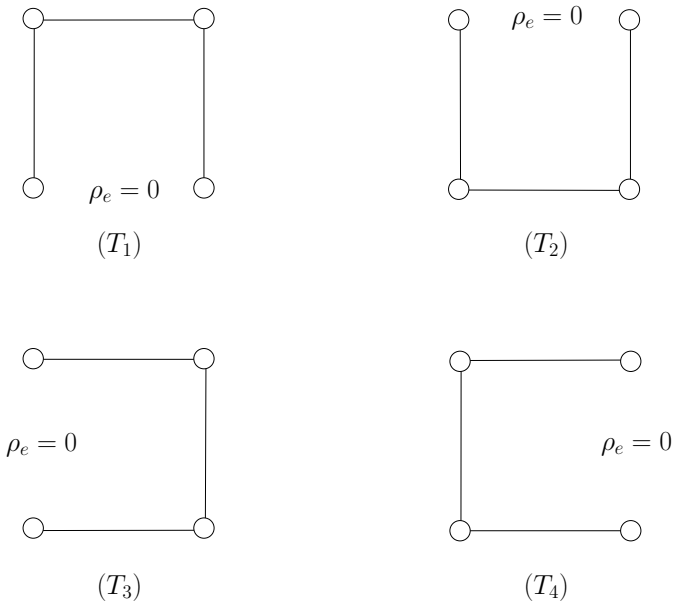

Figure 4.1: A single cycle of length 4 can be expanded to the set $\xi(\mathcal{G}_M)$ consisting of four possible trees with $\rho(T_i) = 1/4$ $(i = 1, 2, 3, 4)$. The EAP of each edge equals $1/4$ when appearing in any T_i and equals 0 otherwise.

mally requires a larger number of iterations and may fail to converge [4]. In [17], the authors developed a novel tree-reweighted BP (TRW-BP) algorithm which improves the convergence behaviour of BP by reweighting certain portions of the factorized graphical representation. Given a Markov random field graph $\mathcal{G}_M$ and the set $\xi(\mathcal{G}_M)$ of all possible trees, a distribution of the trees can be introduced as: $0 \leq \rho(T) \leq 1$ under the constraint $T \in \xi(\mathcal{G}_M)$ with $\sum_{T \in \xi(\mathcal{G}_M)} \rho(T)$. Then, for any distribution $\rho(T)$, the edge appearance probability (EAP) of the edge $e(n, m)$ is defined by:

$$\rho_{nm} = \sum_{T \in \xi(\mathcal{G}_M)} \rho(T) \times \{e(n, m) \in T\}, \tag{4.2}$$

where $\{e(n, m) \in T\}$ is the indicator function whose value equals 1 if the edge $e(n, m) \in T$ and 0 otherwise. To illustrate concepts of possible trees and EAPs, Fig. 4.1 is given where a simple cycle of length 4 is split into 4 possible trees with $\rho(T_i) = 1/4$ $(i = 1, 2, 3, 4)$. In [17], the beliefs which are an approximation of the *a posteriori* marginals $p(x_n|\boldsymbol{y})$ are given by:

$$b_n(x_n) \propto \phi_n(x_n) \Pi_{m \in \mathcal{N}_n} M_{mn}^{\rho_{nm}}(x_n), \tag{4.3}$$

where $\phi_n(x_n)$ denotes the *potential* function corresponding to $p(y_n|x_n)$ for the application of error control coding, and M_{mn} denotes the message from variable x_m to a neighbouring

variable x_n. When $\mathcal{G}_M$ is a tree, $\rho_{nm} = 1$ for all edges $e(n, m)$; when $\mathcal{G}_M$ contains cycles, $\rho_{nm} < 1$ for at least one edge [17].

However, the TRW-BP algorithm only considers a factorized graph with pairwise interactions, and is not suitable for distributed inference problems since it optimizes the reweighting parameters over spanning trees. These issues have been addressed by the URW-BP algorithm reported in [20], which extends the pairwise factorizations of TRW-BP to hypergraphs, and replaces a series of globally optimized parameters with a simple constant. With a small number of decoding iterations, the URW-BP algorithm has been verified to outperform the standard BP algorithm for regular LDPC codes that possess a roughly uniform structure [84]. However, how to choose the optimal ρ is still an open issue.

4.4 Proposed VFAP-BP Decoding Algorithm

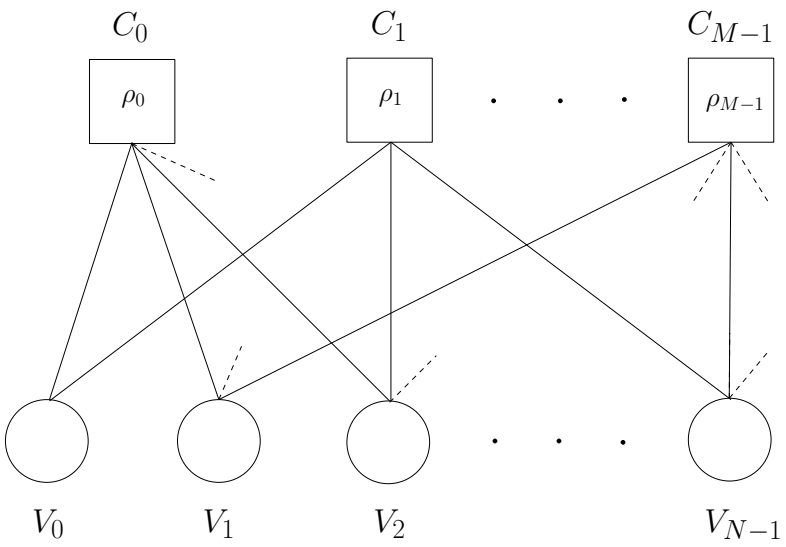

Figure 4.2: The graphical model depicts BP decoding algorithms for LDPC codes, where $\rho_i(i = 0, 1, \ldots, M-1) = 1$ corresponding to the standard BP, $\rho_i(i = 0, 1, \ldots, M-1) = \rho_u$ corresponds to the URW-BP, and $\rho_i(i = 0, 1, \ldots, M-1) = \rho_v$ or 1 depending on a variable condition corresponds to the proposed VFAP-BP.

This section presents the proposed VFAP-BP algorithm, in which we devise a simple criterion for determining the reweighting parameters to improve decoding performance

with respect to both regular and irregular LDPC codes. The idea behind the proposed algorithm is inspired by the fact that the existence of short cycles creates the statistical dependency among the incoming messages being exchanged by nodes, such that the outgoing messages inaccurately have high reliability, or equivalently low quality. This problem is often referred to as "overconfidence" or "overestimation" [17], [20]. As shown in Fig. 4.2, the URW-BP tackles "overconfidence" by assigning a uniform reweighting parameter ρ_u to each parity-check node, resulting in less concentrated and more robust beliefs [20]. On the other hand, it is well known that not all short cycles are equally detrimental with respect to decoding performance. Specifically, check nodes having a large number of short cycles are more likely to form clusters of small cycles, which significantly obstruct the convergence of the BP algorithm within limited iterations [14]. By assigning various FAP values $\rho_i(i = 0, 1, \ldots, M-1)$ in Fig. 4.2, the proposed VFAP-BP algorithm takes advantage of the reweighting strategy as well as the knowledge of the structure of short cycles. According to [20], URW-BP and the optimized TRW-BP are equivalent, as the factor graph $\mathcal{G}$ has a symmetric factorization that refers to codes with regular design. Otherwise, a uniform choice of ρ does not guarantee to improve the convergence of the BP algorithm. On the other hand, a symmetric factor graph is not required for the proposed VFAP-BP algorithm since it adjusts the reweighting parameter based on the knowledge of short cycles, rather than on the factorization of the graph. For this reason, it is also suitable for LDPC codes with irregular designs. In the following, we briefly explain the algorithm that we employ to count short cycles in the factor graph, then introduce the message passing rules and the VFAP-BP decoding algorithm flow.

Given a graphical model with many nodes, counting short cycles exactly in an arbitrary graph seems to be computationally impossible. However, the cycle counting algorithm [15], briefly explained in Chapter 2, transforms the problem of counting cycles into one of counting so-called lollipop walks through matrix multiplications. Note that the counting cycle algorithm in [16] can also be applied and works more efficiently when the sparse graph becomes larger. As a consequence, resorting to either algorithm provides knowledge of the girth g in the factor graph and the number of length-g cycles with respect to every check node $C_i(i = 0, 1, \ldots, M-1)$. In this work, we focus on the value of g, $s_i(i = 0, 1, \ldots, M-1)$, the number of length-$g$ cycles passing a check node C_i, and μ_g, the average number of length-g cycles passing a check node. In a similar way to [17] and [20], the reweighting vector $\rho_i = [\rho_0, \rho_1, \ldots, \rho_{M-1}]$ consists of variable FAP, which

originally describe the probabilities of any check node appearing in a potential spanning tree. As shown in Fig. 4.2, every check node C_i is assigned to a FAP value such that the outgoing messages from a check node are either unchanged or partially reweighted. This depends on whether the outgoing messages from a check node contribute to the extrinsic message passing or not . A check node obstructs convergence or leads to low-quality beliefs, due to creating dependency within the cluster of short cycles. As a result, two cases can be distinguished by a simple criterion: if $s_i < \mu_g$ the check node C_i is regarded as constructive, then $\rho_i = 1$; otherwise this check node is determined as a destructive node and we have $\rho_i = \rho_v$, where $\rho_v = 2/\bar{n_D}$, and $\bar{n_D}$ is the average connectivity for N variable nodes, which is computed as:

$$\bar{n_D} = \frac{1}{\int_0^1 \upsilon(x)dx} = \frac{M}{N\int_0^1 \nu(x)dx}, \tag{4.4}$$

where $\upsilon(x)$ and $\nu(x)$ are the distributions of variable nodes and check nodes. Note that $\rho_v = 2/\bar{n_D}$ at initialization is an approximation of the optimized FAP value according to [17].

The message passing rules of the proposed VFAP-BP algorithm are similar to those derived in [20] for the URW-BP algorithm. We denote the beliefs by LLRs, and these are initialized by $L(x_j) = \log \frac{p(y_j|x_j=1)}{p(y_j|x_j=0)} = 2\frac{y_j}{\sigma^2}$ for an AWGN channel, where σ^2 is the noise variance. The message sent from V_j to C_i is given by:

$$\Psi_{ji} = L(x_j) + \sum_{i' \in \mathcal{N}(j)\backslash i} \rho_{i'} \Lambda_{i'j} - (1-\rho_i)\Lambda_{ij}, \tag{4.5}$$

where $i' \in \mathcal{N}(j)\backslash i$ is the neighboring set of check nodes of V_j except C_i. The quantity Λ_{ij} denotes messages sent from C_i to V_j in the previous iteration, then for all check nodes C_i for $(i = 0, 1, \ldots, M-1)$ we update Λ_{ij} as:

$$\Lambda_{ij} = 2\tanh^{-1}\Big(\prod_{j' \in \mathcal{N}(i)\backslash j} \tanh\frac{\Psi_{j'i}}{2}\Big), \tag{4.6}$$

where '$\tanh(\cdot)$' denotes the hyperbolic tangent function. Finally, we have the belief $b(x_j)$ with respect to x_j described by:

$$b(x_j) = L(x_j) + \sum_{i \in \mathcal{N}(j)} \rho_i \Lambda_{ij}. \tag{4.7}$$

Using the above message passing rules, the proposed VFAP-BP decoding algorithm is depicted in Table 4.1. As an improvement to the URW-BP, the proposed VFAP-BP

Table 4.1: The Algorithm Flow of the VFAP-BP Algorithm

Initialization:

1: Find the girth g and the number of length-g cycles s_i passing through the check node C_i;

2: Determine variable FAPs for each check node: if $s_i < \mu_g$ $\rho_i = 1$, otherwise $\rho_i = \rho_v$ where $\rho_v = 2/\bar{n_D}$;

VFAP-BP decoding:

Step 1: Set I_{max} the maximum number of iterations and initialize $L(\boldsymbol{x}) = 2\frac{\boldsymbol{y}}{\sigma^2}$;

Step 2: Update the message passed from variable node V_j to check node C_i using (4.5), where $\Lambda_{i'j}$ and Λ_{ij} are $0s$ at first iteration;

Step 3: Update the message passed from variable node C_i to check node V_j;

Step 4: Update the belief $b(x_j)$ using (4.7) and decide $\hat{\boldsymbol{x}}$;

Step 5: Decoding stops if $\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \mathbf{0}$ or I_{max} is reached, otherwise go back to Step 2.

requires additional complexity of $\mathcal{O}(gN)$ due to the cycle counting algorithm. Nevertheless, the extra complexity is very small when compared to global optimization with complexity of $\mathcal{O}(M^2N)$. Notice that the computation of counting cycles can be further simplified if the algorithm in [16] is applied to larger sparse graphs. More importantly, the proposed algorithm is capable of improving the performance of BP to decode LDPC codes with uniform structures (regular codes) and with non-uniform structures (irregular codes).

4.5 Simulation Results

In this section, we compare the proposed VFAP-BP with the standard BP and URW-BP using simulations. To illustrate the potential application of the proposed algorithm, we have tested a wide range of LDPC codes with different design methods, two of which are MacKay's regular codes [4] and irregular Quasi-cyclic (QC)-LDPC codes selected as standard codes for WiMax 802.16e [85]. The regular code $(3, 6)$ has a block length $N = 1,000$ and rate $R = 0.5$, while the irregular code has block length $N = 576$, rate $R = 0.5$, and degree distributions $\upsilon(x) = 0.21 \times x^5 + 0.33 \times x^2 + 0.46 \times x$ and $\nu(x) = 0.33 \times x^6 + 0.67 \times x^5$. Notice that for the purpose of a fair comparison the optimized ρ_u of URW-BP is acquired from a numerical method, similar to [20], [84], which is normally larger than $2/\bar{n_D}$ ($\rho_u \approx 0.92$ for a regular code while $\rho_u \approx 0.85$ for an irregular code).

In Fig. 4.3, at different SNRs, the convergence behaviours of the proposed VFAP-BP, the URW-BP, and the standard BP algorithms are compared for decoding regular LDPC code within a small number of iterations. The VFAP-BP converges faster than the other algorithms at a SNR of 2 dB but its advantage diminishes at higher SNR values, resulting from the fact that for the standard BP or URW-BP with a uniform FAP the convergence guarantees are strengthened when the noise variance is reduced [86]. Fig. 4.4 reveals the decoding performance of the three algorithms where the VFAP-BP outperforms the others, whereas the performance gain decreases as more iterations are performed. In the case of irregular codes, the proposed VFAP-BP algorithm still works better than the standard BP, while the asymmetric factorization of the irregular graph reduces the performance of URW-BP, as shown in Fig. 4.5. Moreover, Fig. 4.6 demonstrates that VFAP-BP outperforms the standard BP up to 0.5 dB with a maximum of 10 iterations, even though the performance gap narrows when the number of iterations increases.

Compared to the MacKay and QC-LDPC designs, the PEG [21] is a superior construction method with regard to girth conditioning. The following simulation shows whether the proposed VFAP-BP algorithm manages to improve decoding performance when more advanced codes are used. The PEG LDPC codes have a block length of $500 (N = 500)$ and a code rate of $1/2$. For the regular code tested, the variable codes'

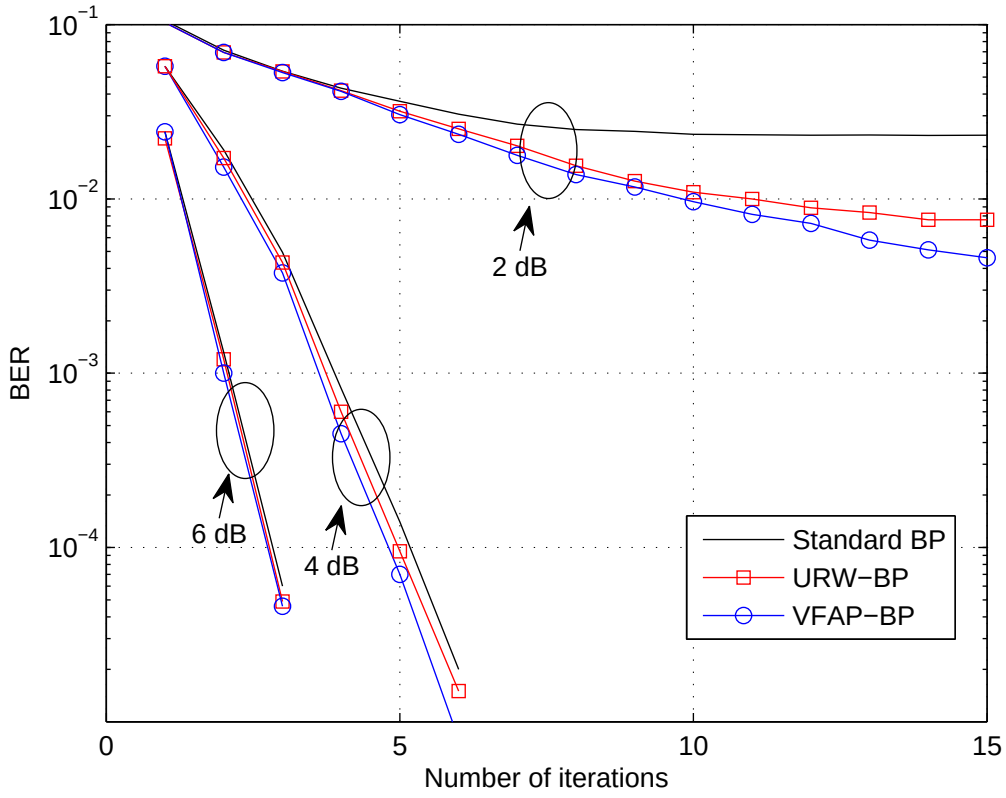

Figure 4.3: Comparison of the convergence behaviour of the URW-BP, VFAP-BP and standard BP algorithms for decoding regular LDPC codes designed by MacKay's method, where SNR equals 2 dB, 4 dB and 6 dB.

degree is $4(\upsilon(x) = x^3)$, the degree of check nodes is $6(\nu(x) = x^5)$ and the average connectivity $\bar{n_D}$ is 4. For irregular code, the degree distribution of variable nodes is $\upsilon(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$, the degree distribution of check nodes is $\nu(x) = x^5$, and the average connectivity $\bar{n_D}$ is 3. After counting the cycles, we find 964 length-6 cycles in the regular graph and 1260 length-8 cycles in the irregular graph.

In Fig. 4.7 the convergent behaviours of the URW-BP, VFAP-BP and standard BP algorithms are compared, in order to illustrate that the proposed algorithm converges faster, particularly in the lower SNR region. Furthermore, Fig. 4.8 reveals the decoding performance of the three algorithms in which the VFAP-BP outperforms the other decoding algorithms regardless of the number of maximum decoding iterations. When decoding irregular codes with asymmetrical graphs, as shown in Fig. 4.9 and in Fig. 4.10, the

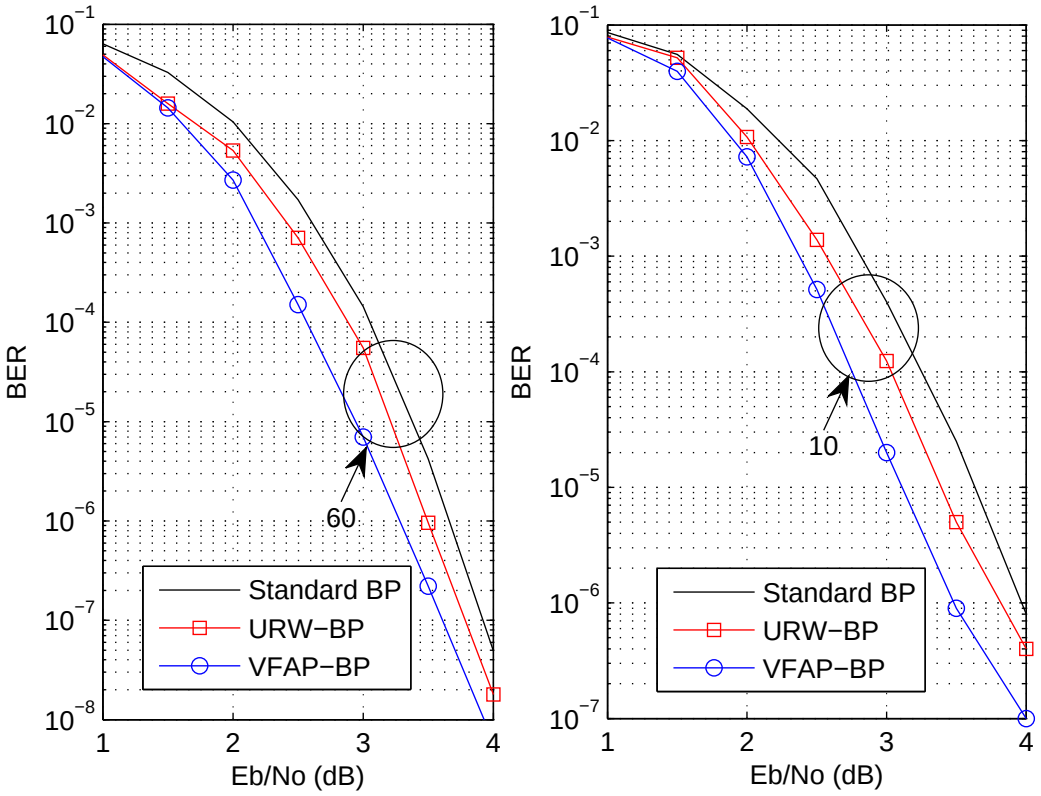

Figure 4.4: Comparison of the BER performance of the VFAP-BP, URW-BP and standard BP algorithms while decoding regular LDPC codes designed by MacKay's method with a maximum of 10 and 60 decoding iterations.

proposed VFAP-BP algorithm still shows better convergence behavior and consistently outperforms the standard BP, but the URW-BP fails to converge at 2 dB and no longer outperforms the standard BP with the maximum number iterations equal to 10 and 60, respectively. As a consequence, for regular and irregular codes with various designs, the proposed VFAP-BP algorithm is able to provide better decoding performance than URW-BP and the standard BP with a limited number of iterations.

4.6 Summary

In this chapter, we have devised a message passing decoding algorithm that employs a reweighting approach and exploits knowledge of the graph structure with short cycles.

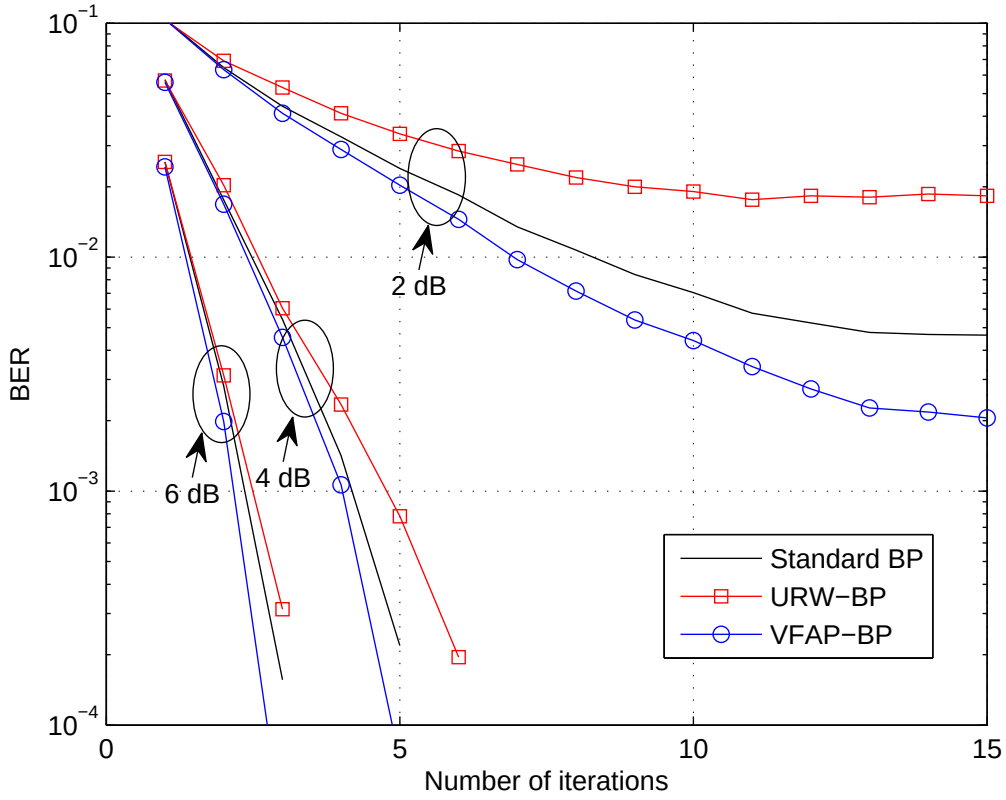

Figure 4.5: Comparison of the convergence behaviour of the URW-BP, VFAP-BP and standard BP algorithms for decoding irregular QC-LDPC codes, where SNR equals 2 dB, 4 dB and 6 dB.

The proposed VFAP-BP algorithm has shown good convergence behavior when compared to the standard BP and the URW-BP algorithms within a limited number of decoding iterations, which is desirable in wireless communication systems with low delay or low latency requirements. Unlike URW-BP, VFAP-BP can also improve the decoding performance over the standard BP when decoding irregular LDPC codes, since it does not require a symmetric factor graph. Nevertheless, the overall performance gain from using the VFAP-BP algorithm is small due to the restriction imposed on the reweighting parameters.

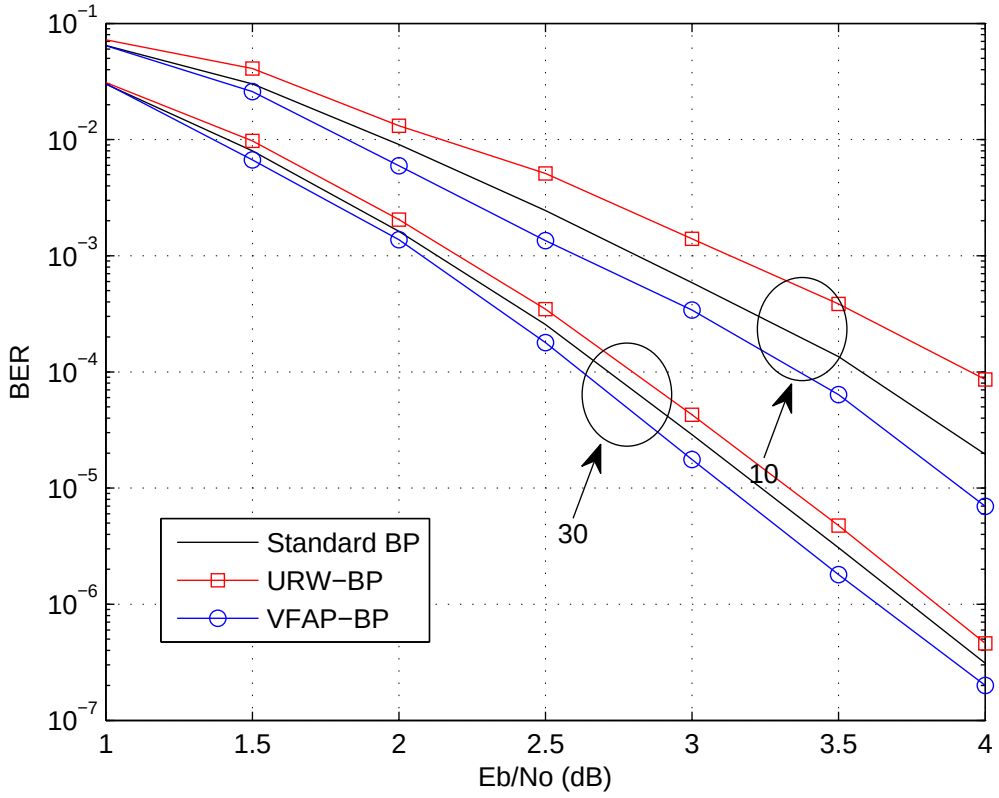

Figure 4.6: Comparison of the BER performance of the VFAP-BP, URW-BP and standard BP algorithms while decoding irregular QC-LDPC codes with a maximum of 10 and 30 decoding iterations.

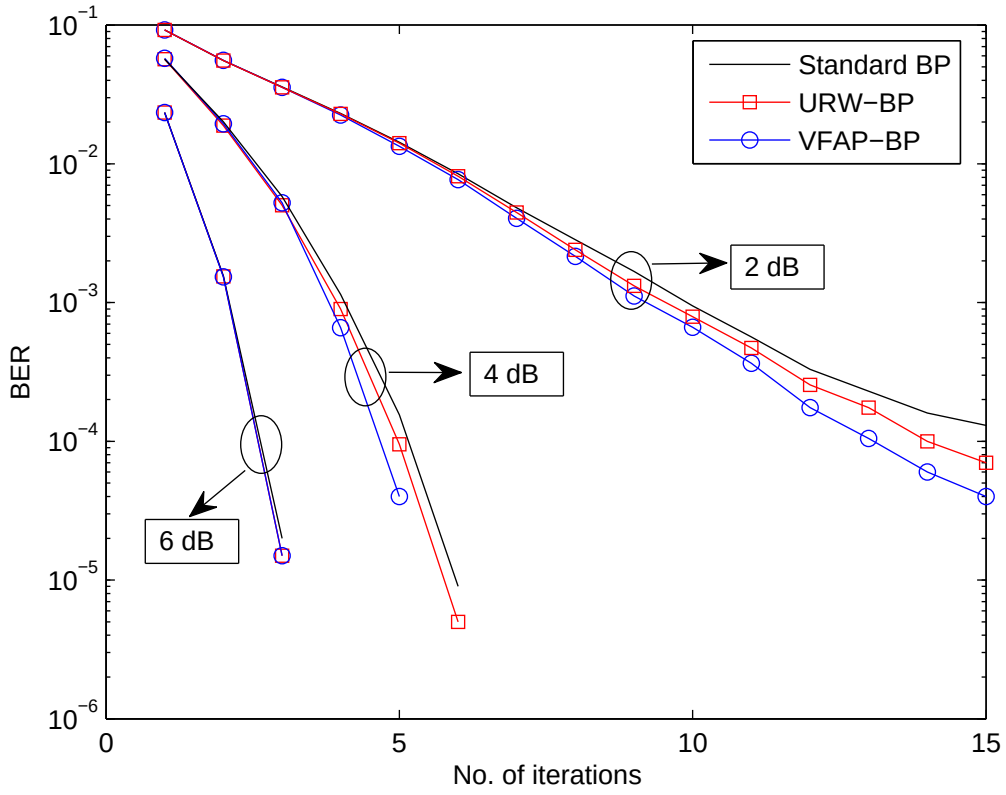

Figure 4.7: Comparison of the convergence behaviour of the URW-BP, VFAP-BP and standard BP algorithms for decoding regular LDPC codes designed by using the PEG algorithm, where SNR equal 2 dB, 4 dB and 6 dB.

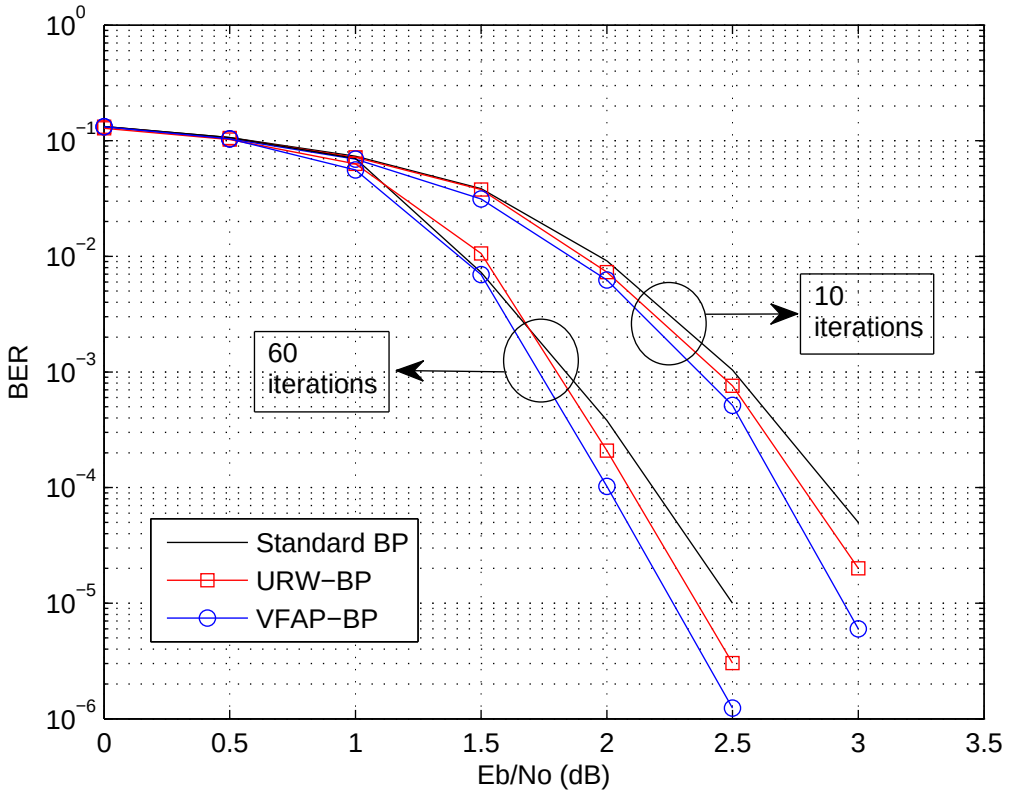

Figure 4.8: Comparison of the BER performance of the VFAP-BP, URW-BP and standard BP algorithms while decoding regular LDPC codes designed by using the PEG algorithm with 10 and 60 maximum decoding iterations.

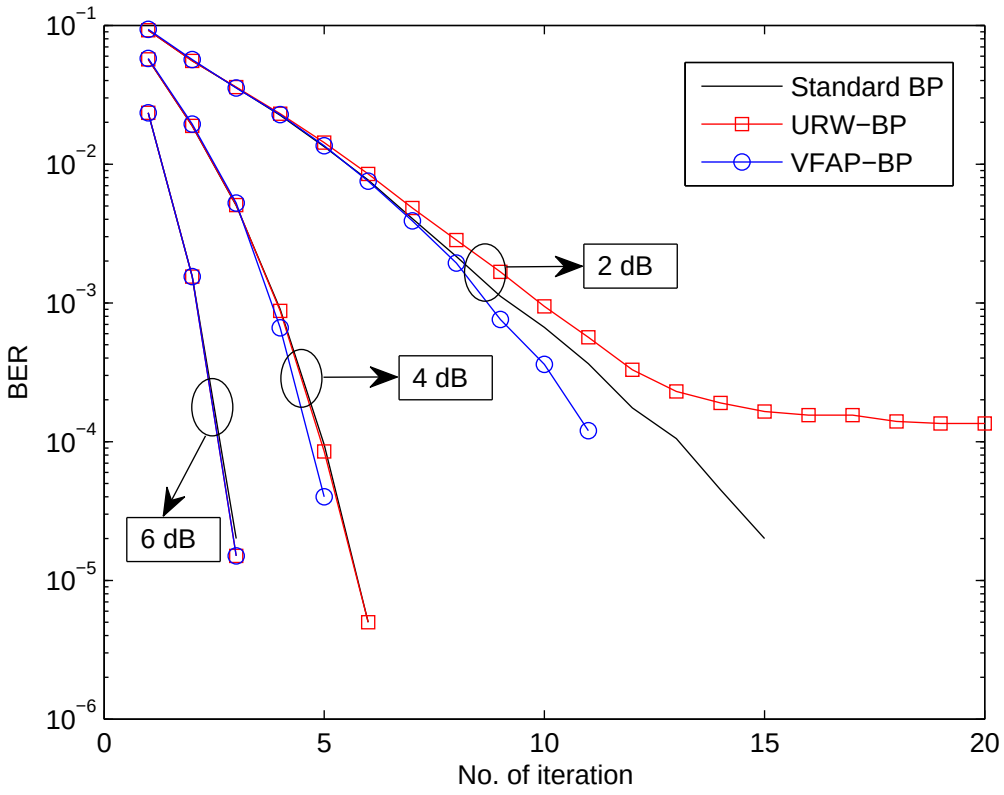

Figure 4.9: Comparison of the convergence behaviour of the URW-BP, VFAP-BP and standard BP algorithms for decoding irregular LDPC codes designed by using the PEG algorithm, where SNR equal 2 dB, 4 dB and 6 dB.

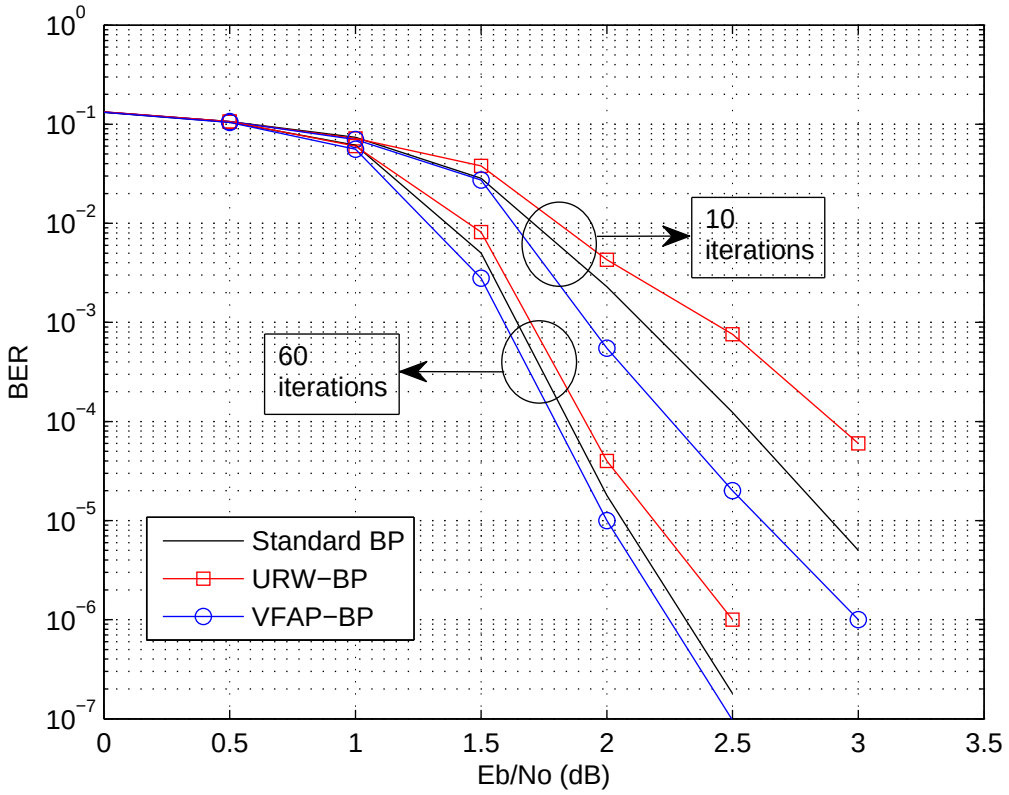

Figure 4.10: Comparison of the BER performance of the VFAP-BP, URW-BP and standard BP algorithms while decoding irregular LDPC codes designed by using the PEG algorithm with 10 and 60 maximum decoding iterations.

Chapter 5

Locally-Optimized Reweighted Belief Propagation for Decoding Finite-Length LDPC Codes

Contents

5.1 Overview

In practice, LDPC codes are decoded using message passing methods. These methods offer good performance, but tend to converge slowly and sometimes fail to converge and to decode the desired codewords correctly. Recently, tree-reweighted message passing methods have been modified to improve the convergence speed at little or no additional complexity cost. This chapter extends the line of work presented in Chapter 4 and pro-

poses a new class of locally optimized reweighting strategies, which are suitable for both regular and irregular LDPC codes. The proposed decoding algorithm first splits the factor graph into subgraphs and subsequently performs a local optimization of reweighting parameters. Simulations show that the proposed decoding algorithm significantly outperforms the standard message passing and existing reweighting techniques.

5.2 Introduction

Recently, a great deal of research has been devoted to the design of LDPC codes with short to moderate block lengths, which correspond to most of the application scenarios of these codes in wireless standards [7].

Decoding is commonly based on iterative message passing methods, allowing local parallel computations. While message passing decoding leads to good performance in terms of BER, it suffers from a number of drawbacks: (i) convergence to a codeword can take many iterations, especially with low SNR; (ii) convergence to a codeword is not guaranteed; (iii) LDPC code design is guided by the decoding algorithm, constraining codes to have large girths. In this context, the occurrence of short cycles and stopping sets makes a significant impact on the performance of LDPC codes, and requires the development of novel decoding strategies that address these problems.

Different approaches have been considered to deal with these issues. The most prominent approach is linear programming (LP) based decoding which, through a relaxation, formulates the decoding problem as an LP problem, and has a maximum likelihood certificate property [87]. LP decoders suffer from high complexity (exponential in the check node degree), unless further relaxations are employed [88]. Another line of investigation, which aims to improve performance while still maintaining the message passing nature of the decoder, is that of tree-reweighted message passing decoding. Based on tree-reweighted belief propagation [17], decoding reverts to a tractable convex optimization problem, iteratively computing beliefs and factor appearance probabilities (FAPs). These concepts were applied to LDPC decoding in [20, 89], where the FAPs were optimized in an off-line procedure subject to additional constraints: in [20], the EAPs were

constrained to be constant, while in [89] as well as in Chapter 4 the FAPs were constrained to take on two possible values. In both cases, gains with respect to standard message passing decoding were observed.

In this chapter, we continue this latter line of research, and explicitly optimize the FAPs offline, without the constraints from [20, 89]. This allows more freedom in the decoding algorithm without additional online computational complexity. We propose a LOW-BP decoding algorithm that first splits the factor graph corresponding to the code into subgraphs and then performs local optimization of the reweighting parameters. The proposed LOW-BP algorithm can mitigate the effects of short cycles and stopping sets in factor graphs by applying a reweighting strategy per subgraph. The LOW-BP algorithm is evaluated for regular and irregular LDPC codes. We have observed significant performance gains over standard message passing decoding. Apart from considering general log-domain BP decoders, another contribution of this chapter is to employ the proposed reweighting and optimization strategy to improve a simplified BP decoding algorithm, i.e. the min-sum decoding algorithm which we briefly introduced in Chapter 2. Min-sum decoding is well known for its low complexity but there is a significant performance degradation due to generating more over-confident extrinsic information than the standard BP or sum-product decoding algorithms [60]. Numerical results show that the application of a LOW-BP algorithm is able largely to enhance the performance of the min-sum decoding algorithm without a reweighting or damping process, which complies with the observations in [84] and [90].

This chapter is structured as follows. Section 5.3 briefly describes the LDPC system model and reviews reweighting strategies and existing algorithms. Section 5.4 is dedicated to a detailed description of the proposed LOW-BP algorithm, whereas Section 5.5 presents and discusses the simulation results. Section 5.6 draws the conclusions from the work.

5.3 System Model for Decoding LDPC Codes

We consider a rate K/N code, with parity check matrix $\mathbf{H}$, and a corresponding set of codewords $\mathcal{C}$. Note that $\boldsymbol{x} \in \mathcal{C}$ if and only if $\boldsymbol{H}\boldsymbol{x}^T = \mathbf{0}$. Assuming binary phase-shift keying and transmission over an AWGN channel, the received data are described by

$$\boldsymbol{y} = 2\boldsymbol{x} - \mathbf{1} + \boldsymbol{n}, \tag{5.1}$$

where $\boldsymbol{n}$ is a sequence of N i.i.d. AWGN samples with variance σ^2, and $\boldsymbol{x} \in \mathcal{C}$ is the transmitted codeword. Given $\boldsymbol{y}$, the aim of the iterative decoder is to recover $\boldsymbol{x}$ in an iterative fashion until either $\boldsymbol{H}\boldsymbol{x}^T = \mathbf{0}$ or the maximum number of decoding iterations is reached. Iterative decoding can be interpreted as message passing on a suitable factor graph, and is often implemented using BP, or a variation thereof.

The factor graph corresponding to our model, $\mathcal{G}(V, E)$, includes the check and variable nodes $V = V_c \cup V_s$, as well as a set of edges, $E \subseteq V_c \times V_s$, such that an edge connecting the check node c_i and the variable node s_j exists in the factor graph only if the entry h_{ij} of the parity-check matrix $\boldsymbol{H}$ equals 1. The decoding process can be interpreted as finding $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y})$. Using Bayes' rule, this a posteriori distribution becomes

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}. \tag{5.2}$$

For any graphical model, such as a Markov field or a factor graph, we assume (5.2) can be factorized as:

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \prod_{n=1}^{N} \phi_n(x_n) \prod_{m=1}^{M} \psi_m(x_{C_m}), \tag{5.3}$$

where $\phi_n(x_n)$ often refers to the *potential* function, which is related to the marginal or conditional distribution of the variable $\boldsymbol{x}$, which in turn corresponds to $p(y_n|x_n)$ in our case, and $\psi_m(x_{C_m})$ is called the *compatibility* function defining the interaction in a clique or factor, for LDPC codes, which corresponds to an indicator function $\{\sum_{n \in C_m} x_n = 0\}$ within $C_m (m = 1, 2, \ldots, M)$ the m-th check [13, 17].

5.3.1 Algorithmic Reweighting Strategies and Variations

When a factor graph contains short cycles, the standard BP algorithm normally requires a larger number of iterations but may still fail to converge. To tackle the issue of non-

convergence, Wainwright et al. [17] presented the tree-reweighted (TRW)-BP algorithm for cyclic graphs, which aims to impose tighter upper bounds on the log-partition function. Alongside the TRW-BP algorithm, a variational approach is provided to derive the reweighting parameters as well as the relevant message passing rules. Given a factor graph the Kullback Leibler divergence [24] between the belief distribution $b(\boldsymbol{x})$ and $p(\boldsymbol{x}|\boldsymbol{y})$, is defined as:

$$\mathrm{KL}(b||p) = \sum_{\boldsymbol{x}} b(\boldsymbol{x}) \log \frac{b(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})} \geq 0. \tag{5.4}$$

Combining the above equation with (5.3) and performing some straightforward manipulations, we have an inequality as follows:

$$\log p(\boldsymbol{y}) \geq \mathcal{H}(b) + \chi(b), \tag{5.5}$$

in which $\mathcal{H}(b)$ denotes the entropy of the belief and $\chi(b)$ is defined as:

$$\chi(b) = \sum_{n=1}^{N} \sum_{x_n} b_n(x_n) \log \phi_n(x_n) + \sum_{m=1}^{M} \sum_{x_{C_m}} b_{C_m}(x_{C_m}) \log \psi_m(x_{C_m}). \tag{5.6}$$

Notice that if we use C_m to denote the m-th node cluster ($m = 1, 2, \ldots, M$), then $b_{C_m}(x_{C_m})$ denotes joint belief term [24]. In [17], the authors pointed out that (5.4) is valid with equality if and only if $b(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$. Since the fixed points of the BP algorithm correspond to the stationary points of Bethe's free energy [55], the entropy term in (5.4) can be replaced by the so-called Bethe entropy with the reweighting factors being:

$$\mathcal{H}_{\mathrm{Bethe}} = \sum_{n=1}^{N} \mathcal{H}(b_n) - \sum_{m=1}^{M} \rho_m \mathcal{I}_{C_m}(b_{C_m}), \tag{5.7}$$

where $b(\cdot)$ denotes the marginal belief, $\mathcal{H}(b_n)$ is the entropy of the belief of the n-th variable and $\mathcal{I}_{C_m}(b_{C_m})$ represents a joint mutual information term [24]. Moreover, $\phi_n(x_n)$ and $\psi_m(x_{C_m})$ are the *potential* function and the *compatibility* function, respectively, which are defined depending on the application. A multi-objective function for optimizing the FAPs $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$, where $M = N - K$, is also given:

$$\log p(\boldsymbol{y}) = \max\{\mathcal{F}(\boldsymbol{b}, \boldsymbol{\rho})\} = \max\{\mathcal{H}_{\mathrm{B}} + \chi(b)\}. \tag{5.8}$$

The optimization with respect to $(\boldsymbol{b}, \boldsymbol{\rho})$ starts with a fixed $\boldsymbol{\rho}^{(k)}$, then solves this for the stationary points of $\mathcal{F}(\boldsymbol{b}, \boldsymbol{\rho}^{(k)})$ via TRW-BP. Next, for a fixed belief vector $\boldsymbol{b}$, minimizing the function $\mathcal{F}(\boldsymbol{b}, \boldsymbol{\rho}^{(k)})$ with respect to $\boldsymbol{\rho}^{(k)}$ results in an updated $\boldsymbol{\rho}^{(k+1)}$. This algorithm keeps running recursively until the belief converges. Observe that standard BP corresponds to

the sub-optimal and generally invalid choice $\boldsymbol{\rho} = \mathbf{1}$. The work reported in [17] is not directly applicable to problems such as the decoding of LDPC codes and only derives message passing rules for graphs with pairwise interactions.

In [20], [84], the URW-BP algorithm extends pairwise factorizations to higher order interactions and reduces a series of globally optimized parameters $\boldsymbol{\rho} \in (0, 1]^M$ to a simple constant $\rho_u \in (0, 1]$. Additionally, the FAPs are generalized to EAPs so that the problem size is significantly reduced. Another reweighting strategy is referred to as VFAP-BP reported in [89] as well as in Chapter 4 that aims to select $\boldsymbol{\rho}$ on the basis of the cycle distribution of the graph. However, neither URW-BP nor VFAP-BP optimizes the values of $\boldsymbol{\rho}$ explicitly.

5.4 Proposed LOW-BP Algorithm for Decoding LDPC Codes

In this section, we describe the LOW-BP algorithm, which explicitly optimizes the reweighting parameter vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$. By allowing optimization over smaller subgraphs, LOW-BP is able to trade off complexity vs. performance. LOW-BP comprises an offline phase, during which, for a fixed SNR and a fixed code, the best choice of $\boldsymbol{\rho}$ is determined, as outlined in Table 5.1. The online phase of LOW-BP occurs during real-time decoding, when optimized $\boldsymbol{\rho}$ is used in the reweighted message passing decoding algorithm.

5.4.1 Offline Phase of LOW-BP

In the offline phase, we transform the factor graph into a set of $T \geq 1$ subgraphs and then locally optimize the reweighting parameter vector $\boldsymbol{\rho}_t$ for each subgraph, where $t = 1, 2, \ldots, T$. Note that when $T > 1$ the dimension of $\boldsymbol{\rho}_t$ depends on the size of the t-th subgraph. The optimization turns out to be significantly less complex when more subgraphs are considered, hence there is a need for a flexible method to decompose the original factor graph into subgraphs. We apply the PEG technique [21] to this end.

Table 5.1: LOW-BP for Decoding LDPC Codes

Offline: subgraphs formation

1: Given an expansion strategy (*disjoint* or *RA*) and $d_{\max}$ which is the maximum expansion level, apply PEG expansion to generate $T \geq 1$ subgraphs;

Offline: optimization of $\boldsymbol{\rho}_t$ for the t-th subgraph

2: Initialize $\boldsymbol{\rho}_t^{(0)}$ to an appropriate value;

3: For each subgraph, utilising training sequence to calculate the beliefs $b(\boldsymbol{x}_t)$ and the mutual information term $\boldsymbol{I}_t = [I_{t,1}, I_{t,2}, \ldots, I_{t,L_t}]$ by using the reweighted message passing rules (5.13)–(5.15);

4: With $b(\boldsymbol{x}_t)$ and $\boldsymbol{I}_t$ obtained from step 3, update $\boldsymbol{\rho}_t^{(r)}$ to $\boldsymbol{\rho}_t^{(r+1)}$ using the conditional gradient method (detailed in the Appendix);

5: Repeat steps 3–4 until $\boldsymbol{\rho}_t$ converges for each subgraph;

Offline: choice of $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$ for decoding

6: For all T subgraphs, collect $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_i, \ldots, \boldsymbol{\rho}_T$. In case of disagreement on a value ρ_m for the m-th check node, choose the value offering the best performance;

Online: real-time decoding

7: Use reweighted message passing decoding (5.13)–(5.15) with optimized $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$ during actual data transmission.

Construction of T Subgraphs

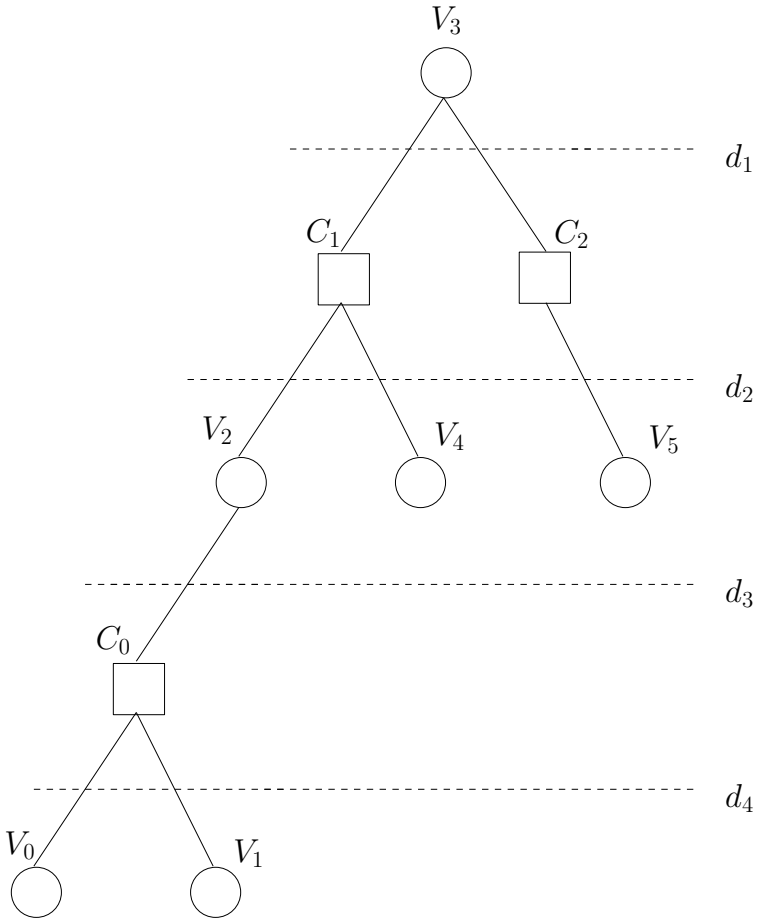

Figure 5.1: Employing the PEG expansion to construct one of the subgraphs originating from Fig. 2.3 in Chapter 2. Note that here variable node V_3 is selected as the root node and $d_{\max} = d_4$.

PEG expands $\mathcal{G}(V, E)$ into T subgraphs $\mathcal{G}_t(V_t, E_t)$, where V_t and E_t are subsets of V and E, respectively. This method is straightforward in use if the LDPC code was designed by PEG, or its variations [81,91], but is not limited to such designs. For instance, given a factor graph designed by Gallager's method we can still conduct graph expansion using the modified PEG algorithm based on its original degree distributions. However, by doing that, it is possible to have some very small or large subgraphs due to inconsistently distributed nodes. So a threshold regarding graph size might be needed to exclude under size or over size subgraphs. We consider a *disjoint* strategy and a *re-appearance (RA)* strategy to apply PEG. The disjoint strategy prohibits duplicates of check nodes in all subgraphs, while the RA strategy allows check nodes to appear more than once over T subgraphs. In general, the number of subgraphs T depends on: (i) a pre-set maximum

Algorithm 1 PEG expansion for the t-th subgraph ($t = 1, 2, \ldots, T$)

while the complementary set V_t is not empty **do**
  for $j = 0 \rightarrow N - 1$ **do**
    for $k = 0 \rightarrow w_{s_j} - 1$ **do**
      if $k = 0$ **then**
        add the first edge(c_i, s_j) denoted by $e^0_{s_j}$ to s_j, such that c_i has the lowest degree under the current subgraph
      else
        keep expanding the subgraph from root s_j and remove (c_i, s_j) from V_t until the maximum level $d_{\max}$ is reached or $\bar{\mathcal{N}}^d_{s_j} \neq \emptyset$ but $\bar{\mathcal{N}}^{d+1}_{s_j} = \emptyset$
      end if
    end for
  end for
end while

expansion level $d_{\max}$, as a large $d_{\max}$ results in a small T but a high probability of the existence of very short cycles within subgraphs; (ii) given $d_{\max}$, whether all the nodes of V are included in the expanded subgraphs. Let us denote the degree of a variable node s_j by w_{s_j}, and define $\mathcal{N}^d_{s_j}$ as the neighborhood of s_j at current expansion level d, as well as $\bar{\mathcal{N}}^d_{s_j}$ being the complement of $\mathcal{N}^d_{s_j}$. To generate the t-th subgraph $\mathcal{G}_t(V_t, E_t)$ based on $\mathcal{G}(V, E)$, the PEG expansion is detailed in Algorithm 1. In the case of the *RA* strategy, V_t, the set of candidate nodes of $\mathcal{G}_t(V_t, E_t)$, is always initialized as V, for each of the T expansions. On the other hand, V_t is the complement set of V_{t-1} if the *disjoint* strategy is applied, so that the size of subgraph $\mathcal{G}_t(V_t, E_t)$ decreases as t increases. Furthermore, some of the subgraphs, such as $\mathcal{G}_T(V_T, E_T)$ or $\mathcal{G}_{T-1}(V_{T-1}, E_{T-1})$, may be acyclic (i.e., a tree), with corresponding reweighting factors $\boldsymbol{\rho}_t = \mathbf{1}$, which complies with the observations in [17], [20]. Compared to the greedy search algorithm in [21, 91], our PEG-based expansion stops as soon as every member of V_t has been visited. The number of edges incident to s_j might be less than w_{s_j} as some short cycles are excluded from the subgraphs so as to guarantee that the local girth of each subgraph is always larger than the global girth of the original graph.

Optimization of FAPs

After obtaining T subgraphs, we introduce $\boldsymbol{L} = [L_1, L_2, \ldots, L_T]$, where L_t is the number of check nodes (possibly with duplicates) in the t-th subgraph. Note that $\sum_t L_t = M$ if a disjoint expansion is used, while $\sum_t L_t > M$ if the RA expansion is employed. Observe also that when $T = 1$, then $L = M$. With the t-th subgraph, we optimize the associated FAPs $\boldsymbol{\rho}_t = [\rho_{t,1}, \rho_{t,2}, \ldots, \rho_{t,L_t}]$ using the optimization method for TRW-BP [17], modified to higher-order interactions, with the corresponding message passing rules from [84]. The optimization problem is solved recursively as follows, starting from the initial values $\boldsymbol{\rho}_t^{(0)}$ for each of the T subgraphs:

1. For all T subgraphs in parallel, for fixed $\boldsymbol{\rho}_t^{(r)}$, use message passing to calculate the beliefs $b(\boldsymbol{x}_t)$ and the mutual information term $\boldsymbol{I}_t = [I_{t,1}, I_{t,2}, \ldots, I_{t,L_t}]$, provided that there are $L_t \leq M$ check nodes in the t-th subgraph.

2. For all T subgraphs in parallel, given $\{\boldsymbol{I}_t\}_{t=1}^T$, use the conditional gradient method to update, for all t, $\boldsymbol{\rho}_t^{(r)}$ to $\boldsymbol{\rho}_t^{(r+1)}$, then go back to step 1.

According to (5.8), for a specific subgraph $\mathcal{G}_t(V_t, E_t)$ our goal is to minimize the following objective function with respect to the column vector $\boldsymbol{\rho}_t$:

$$\begin{aligned}\mathcal{F}(\boldsymbol{b}, \boldsymbol{\rho}) = &\sum_{n=1}^{N} \mathcal{H}(b_n) - \sum_{m=1}^{M} \rho_m \mathcal{I}_{C_m}(b_{C_m}) \\ &+ \sum_{n=1}^{N} \sum_{x_n} b_n(x_n) \log \phi_n(x_n) \\ &+ \sum_{m=1}^{M} \sum_{x_{C_m}} b_{C_m}(x_{C_m}) \log \psi_m(x_{C_m}).\end{aligned} \tag{5.9}$$

Dropping terms that do not depend on $\boldsymbol{\rho}_t$, we find the following optimization problem with $\boldsymbol{I}_t = [I_{t,1}\ I_{t,2} \ldots I_{t,L_t}]^T$, where $(\cdot)^T$ denotes the transpose:

$$\begin{aligned}\text{minimize} \quad & -\boldsymbol{\rho}_t^T \boldsymbol{I}_t \\ \text{s.t.} \quad & \boldsymbol{\rho}_t \in \mathbb{T}\big(\mathcal{G}_t(V_t, E_t)\big),\end{aligned}$$

where $\mathbb{T}\big(\mathcal{G}_t(V_t, E_t)\big)$ is the set of all valid FAPs over the subgraph $\mathcal{G}_t(V_t, E_t)$ and $I_{t,l}$ is a mutual information term, which depends on $\boldsymbol{\rho}_t^{(r)}$, the previous value of $\boldsymbol{\rho}_t$. We will

denote the objective function by $f(\boldsymbol{\rho}_t) = -\boldsymbol{\rho}_t^T \boldsymbol{I}_t$ and use the conditional gradient method to update $\boldsymbol{\rho}_t$, in a similar way to [17]. In the conditional gradient method, we first linearize the objective around the current value $\boldsymbol{\rho}_t^{(r)}$:

$$f_{\text{lin}}(\boldsymbol{\rho}_t) = f(\boldsymbol{\rho}_t^{(r)}) + \nabla_{\boldsymbol{\rho}_t}^T f(\boldsymbol{\rho}_t^{(r)})(\boldsymbol{\rho}_t - \boldsymbol{\rho}_t^{(r)}), \tag{5.10}$$

in which $\nabla_{\boldsymbol{\rho}_t} f(\boldsymbol{\rho}_t^{(r)}) = -\boldsymbol{I}_t$. Secondly, we minimize $f_{\text{lin}}(\boldsymbol{\rho}_t)$ with respect to $\boldsymbol{\rho}_t$, denoting the minimizer by $\boldsymbol{\rho}_t^*$ and $z_t^{(r+1)} = \max(f_{\text{lin}}(\boldsymbol{\rho}_t^*), z_t^{(r)})$, where $z_t^0 = -\infty$. Finally, $\boldsymbol{\rho}_t^{(r)}$ is updated to $\boldsymbol{\rho}_t^{(r+1)}$ as

$$\boldsymbol{\rho}_t^{(r+1)} = \boldsymbol{\rho}_t^{(r)} + \alpha(\boldsymbol{\rho}_t^* - \boldsymbol{\rho}_t^{(r)}), \tag{5.11}$$

in which α is chosen as

$$\arg \min_{\alpha \in [0,1]} f(\boldsymbol{\rho}_t^{(r)} + \alpha(\boldsymbol{\rho}_t^* - \boldsymbol{\rho}_t^{(r)})). \tag{5.12}$$

At every iteration, $f(\boldsymbol{\rho}_t^{(r)})$ is an upper bound on the optimized objective, while $z_t^{(r+1)}$ is a lower bound.

When applying the RA strategy, it is possible for a check node to have a non-unique FAP. In that case, we try different FAP values corresponding to one specific check node while fixing FAPs for all other check nodes then choose the FAP that gives the best BER performance through simulations. For clarity, the two steps above are referred to as recursion. The optimization of the FAPs runs for as many recursions as are necessary until each $\boldsymbol{\rho}_t$ converges, in order to acquire the optimal reweighting vector.

5.4.2 Online Phase of LOW-BP

Once the optimized values of $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$ are found, actual data transmission can commence. For completeness, we briefly review the message passing rules applied here, and refer the reader to [84] for more details. All messages are represented in LLRs. For an AWGN channel with noise variance σ^2, the message from the n-th variable node s_n to the m-th check node c_m is given by

$$\Psi_{nm} = \lambda_{\text{ch},n} + \sum_{m' \in \mathcal{N}(n) \backslash m} \rho_{m'} \Lambda_{m'n} - (1 - \rho_m)\Lambda_{mn}, \tag{5.13}$$

where $\lambda_{\text{ch},n} = \log(p(y_n|x_n = 1)/p(y_n|x_n = 0)) = 2y_n/\sigma^2$, $m' \in \mathcal{N}(n)\backslash m$ is the neighbouring set of check nodes of s_n except c_m. The quantity Λ_{mn} denotes messages

sent from c_m to s_n in a previous decoding iteration, then for check nodes c_m we update Λ_{mn} as:

$$\Lambda_{mn} = f_{\boxplus}\big(\{\rho_m \Psi_{nm'}\}_{m' \in \mathcal{N}(n) \backslash m}\big) - (1 - \rho_m)\Psi_{nm}, \tag{5.14}$$

where $f_{\boxplus}(\cdot)$ denotes the standard BP message passing rule to compute an LLR message from check node c_m to variable node s_n. The function $f_{\boxplus}(\cdot)$ can be implemented by using the well-known hyperbolic tangent expressions [89], or the numerically more stable Jacobian logarithm, [13] and [92]. Upon convergence, we have belief $\lambda_{\text{belief},n}$ with respect to x_n given by

$$\lambda_{\text{belief},n} = \lambda_{\text{ch},n} + \sum_{m \in \mathcal{N}(n)} \rho_m \Lambda_{mn}. \tag{5.15}$$

It should be noted that in (5.13)–(5.15), the standard BP or sum-product algorithm corresponds to $\rho_m = 1, \forall m$. The receiver utilizes the above message passing rules and does not update $\boldsymbol{\rho}$ as long as the channel conditions are unchanged.

If the min-sum decoding algorithm is used, the message sent from check nodes c_m to variable nodes s_n (5.14) should be modified as:

$$\Lambda_{mn} = \prod_{n' \in N(m)/n} \text{sign}(\rho_m \Psi_{mn'}) \min(|\rho_m \Psi_{mn'}|). \tag{5.16}$$

As the hyperbolic function in the standard BP or sum-product algorithm is substituted by the min-operations, the overall complexity of the min-sum algorithm is largely reduced due to the involvement of additions and min-operations only. However, it has been proven that the complexity reduction of the min-sum algorithm is realized at the expense of performance degradation [92].

5.5 Simulation Results

In this section, we show the numerical results obtained from applying the proposed LOW-BP algorithm to the decoding of regular and irregular LDPC codes with short block lengths, over the AWGN channel. Regular code has block length $N = 500$ and rate $R = 1/2$, with constant column weight $w_s = 4$ and row weight $w_c = 6$. Irregular code has the same block length and rate, but a variable node degree distribution $\upsilon(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$ and a constant check node degree

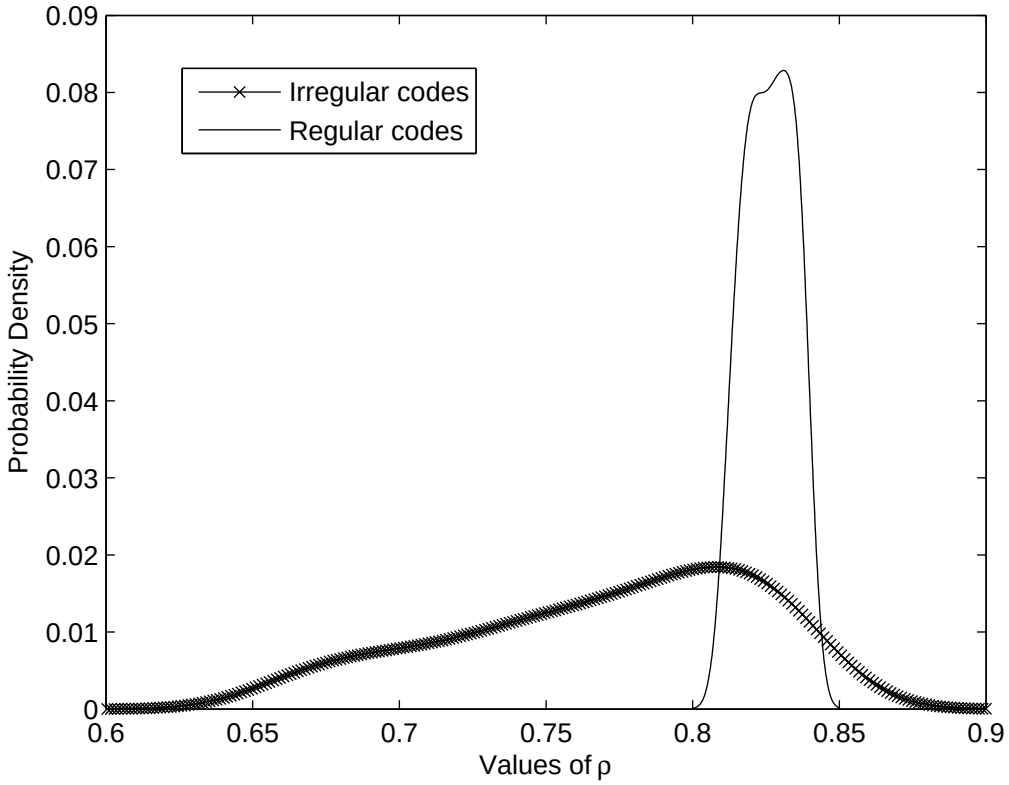

Figure 5.2: Histograms of the $\boldsymbol{\rho}$ values for regular codes and irregular codes at an SNR of 2 dB. The $\boldsymbol{\rho}$ is derived by using LOW-BP optimization with disjoint selection and run until convergence.

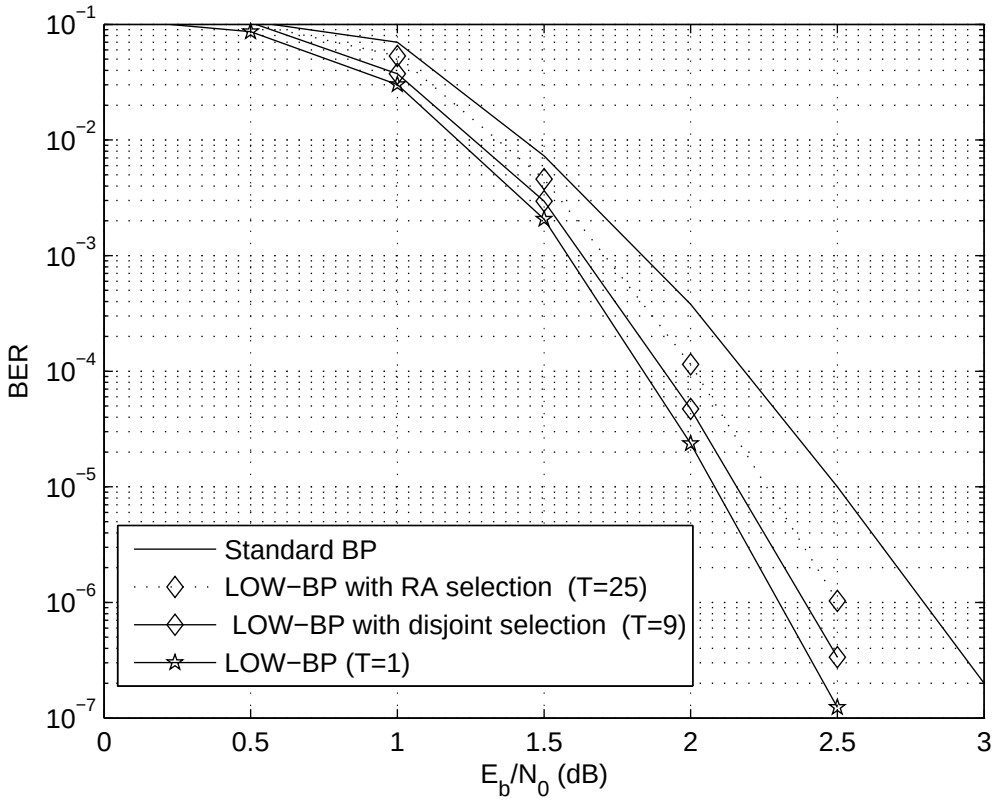

Figure 5.3: Comparison of decoding performance using the proposed LOW-BP algorithm with various numbers of subgraphs T for regular codes.

of 5. For the sake of numerical stability and data storage, all messages are represented as LLRs, and the Jacobian logarithm [13, 92] is used to compute the messages passed from check nodes to variable nodes. In the offline phase of LOW-BP, $1,000$ codewords known to the receiver are transmitted so as to optimize $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$. For all decoding algorithms, we allowed up to 60 decoding iterations in the online phase.

Fig. 5.2 illustrates the distribution of the reweighting parameters for regular codes and irregular codes, at an SNR of 2 dB. It is clear that the optimized $\boldsymbol{\rho}$ of irregular codes is widely distributed over the range of $[0.6, 0.9]$, while the $\boldsymbol{\rho}$-distribution for regular codes is more concentrated within a smaller range $[0.8, 0.85]$. This observation is consistent with the findings in [17, 20], which state that for symmetric graphs, the optimal reweighting parameters should be more or less uniform.

In Fig. 5.3 and Fig. 5.4, the BER performance with different values of T, with RA and disjoint selection, are compared to the standard BP algorithm, for regular and irregular

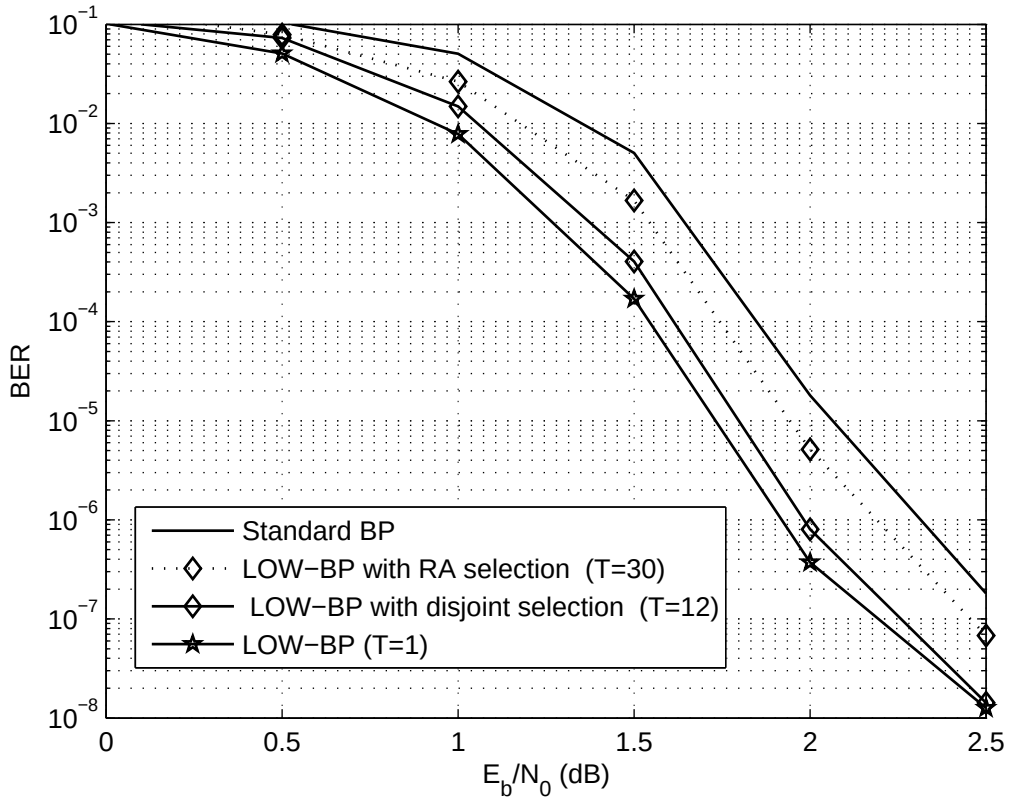

Figure 5.4: Comparison of decoding performance using the proposed LOW-BP algorithm with various numbers of subgraphs T for irregular codes.

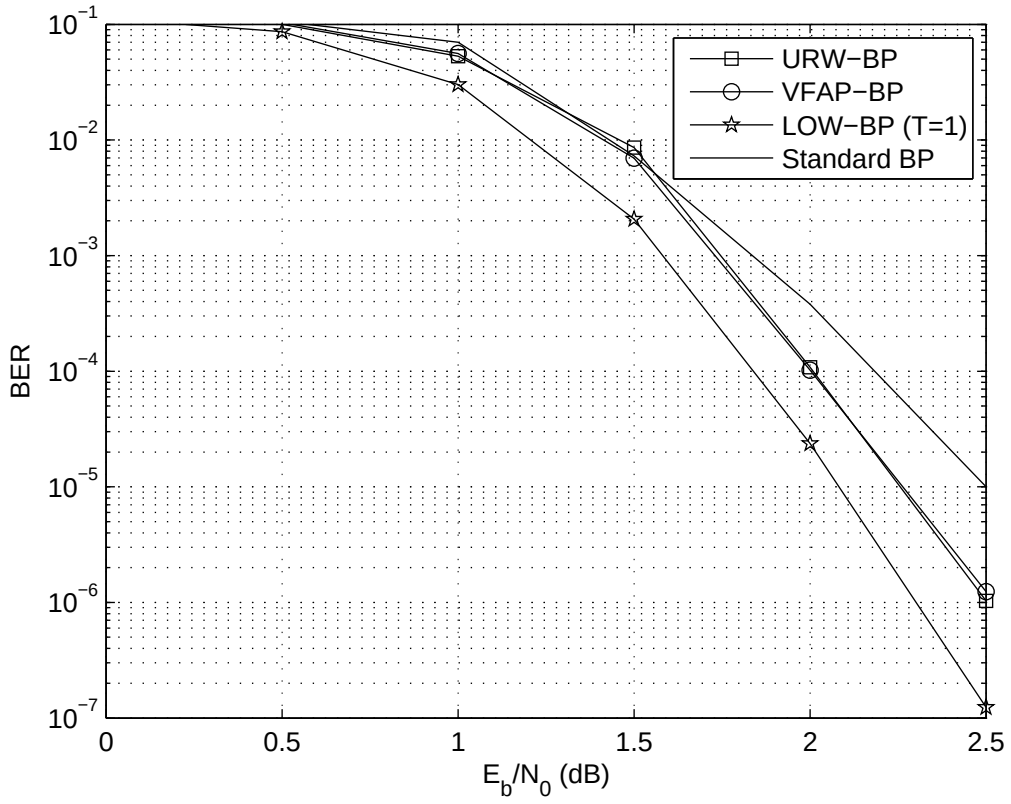

Figure 5.5: Comparison of the performance of BP, URW-BP, VFAP-BP and the proposed LOW-BP for decoding regular codes.

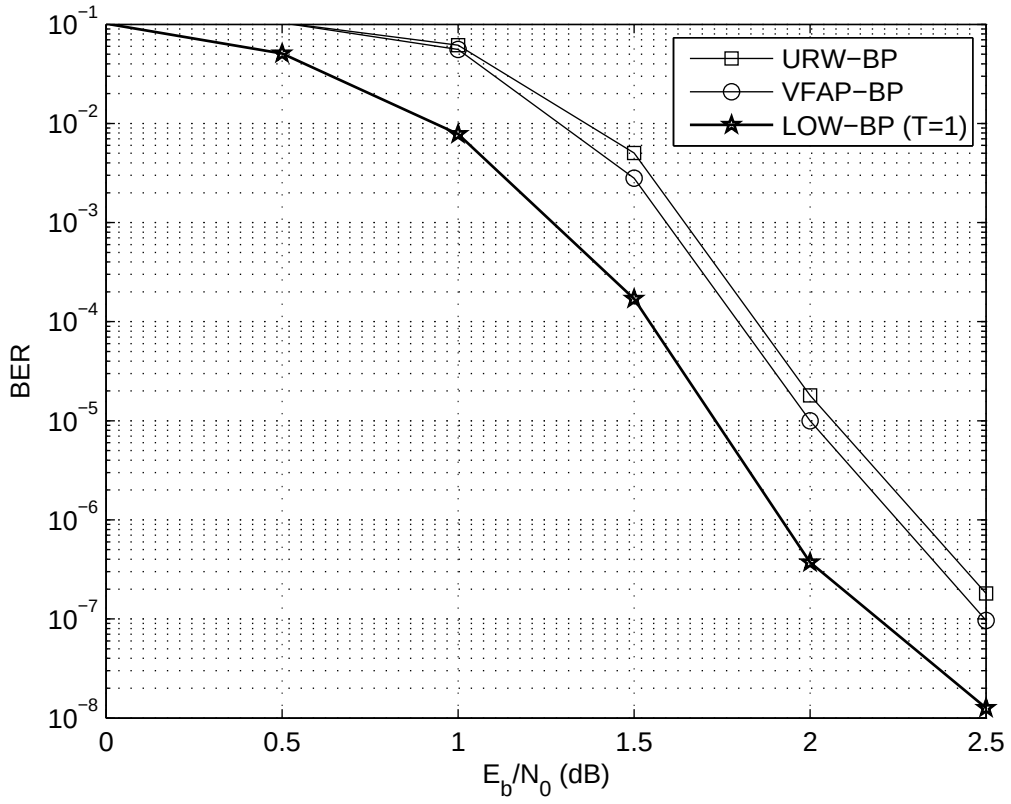

Figure 5.6: Comparison of the performance of BP (same results as URW-BP), URW-BP, VFAP-BP and the proposed LOW-BP for decoding irregular codes.

codes, respectively. We observe a performance gain of up to 0.4 dB over the standard BP algorithm by using the proposed LOW-BP method. For regular code, all check nodes are visited once in $T = 9$ subgraphs with the disjoint selection, while with RA selection $T = 25$ subgraphs are generated (maximum 60 recursions), where some check nodes are revisited. For irregular code, the disjoint selection generates $T = 12$, meanwhile the RA selection gives $T = 30$ (maximum 100 recursions). When using disjoint selection, $\boldsymbol{\rho}$ converges to a set of stable values after a number of recursions that varies from one subgraph to another. When using disjoint selection the graph expansion stops until all the check nodes have been included, so the value of T is fixed. While for the RA selection, the value of T has a minimum at which all the check nodes should be covered and can also be increased if more subgraphs are required. Notice that in both figures, $T = 1$ is a special case that corresponds to Wainwright et al.'s optimal solution from [17]. For $T = 1$, to improve convergence in the offline phase, we initialized $\boldsymbol{\rho}$ from URW-BP [20] for regular code and from VFAP-BP [89] for irregular code. Normally, around 800 offline recursions are required to converge for regular code and $2,700$ offline recursions for irregular code. In practice, we found that $T = 1$ makes for very slow convergence of $\boldsymbol{\rho}$, especially for large graphs.

A comparison with existing reweighted methods is shown in Fig. 5.5 and Fig. 5.6, for regular and irregular codes, respectively. The algorithms considered are URW-BP from [20], VFAP-BP from [89], and the proposed LOW-BP algorithm (with $T = 1$). For regular code, we observe that URW-BP and VFAP-BP outperform standard BP. LOW-BP is able to provide further improvements. For irregular code, the optimal constant value of the FAP in URW-BP is $\rho = 1$, so that BP and URW-BP coincide. VFAP-BP provides a small performance gain, while LOW-BP again outperforms BP by up to 0.4 dB. We clearly see that explicit optimization of $\boldsymbol{\rho}$ leads to non-trivial performance gains.

Additionally, we tested the proposed LOW-BP decoding algorithm and its counterparts with the min-sum algorithm. Using the same irregular code, the histograms of $\boldsymbol{\rho}$ values, from the interval $[0.4, 0.72]$, are shown in Fig. 5.7. For comparison purposes, the same curve in Fig. 5.2 is included. Notice that in this situation the distribution of reweighting parameters $\boldsymbol{\rho}$ is even more scattered when compared with the sum-product algorithm. The reason why the min-sum decoding algorithm has smaller $\boldsymbol{\rho}$ values is because it overestimates the likelihood of variable $\boldsymbol{x}$, such that the reweighting parameters need to be

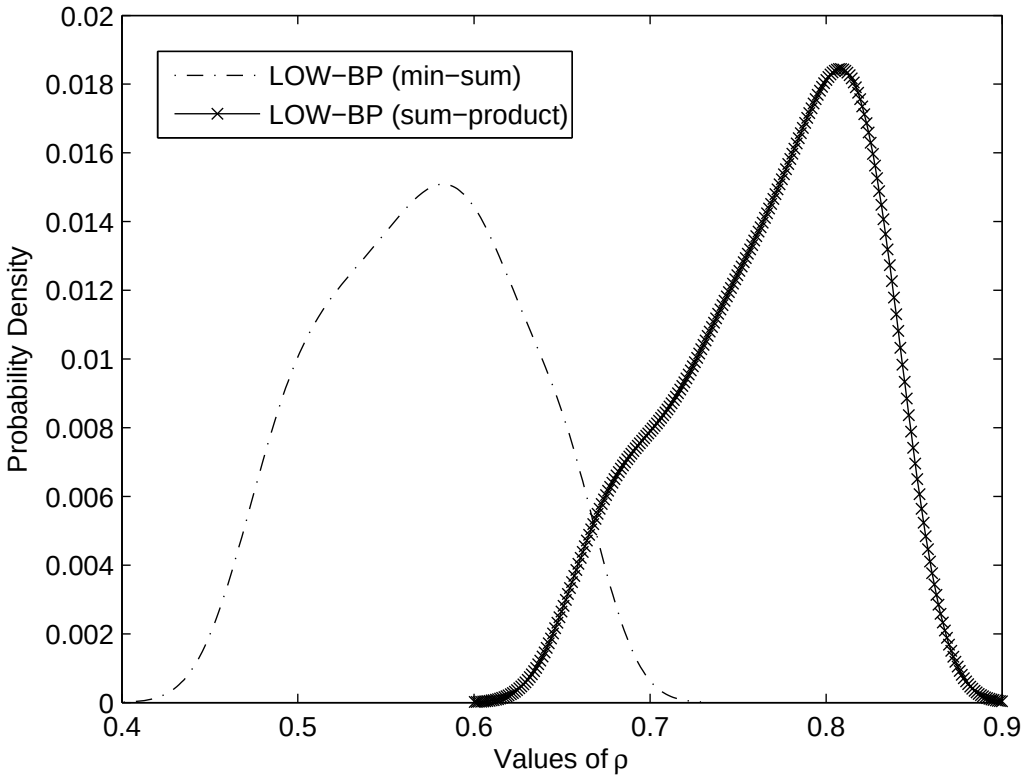

Figure 5.7: Histograms of the ρ values for irregular codes based on the min-sum decoding algorithm at an SNR of 2 dB. The ρ is derived by using LOW-BP optimization with disjoint selection and run until convergence.

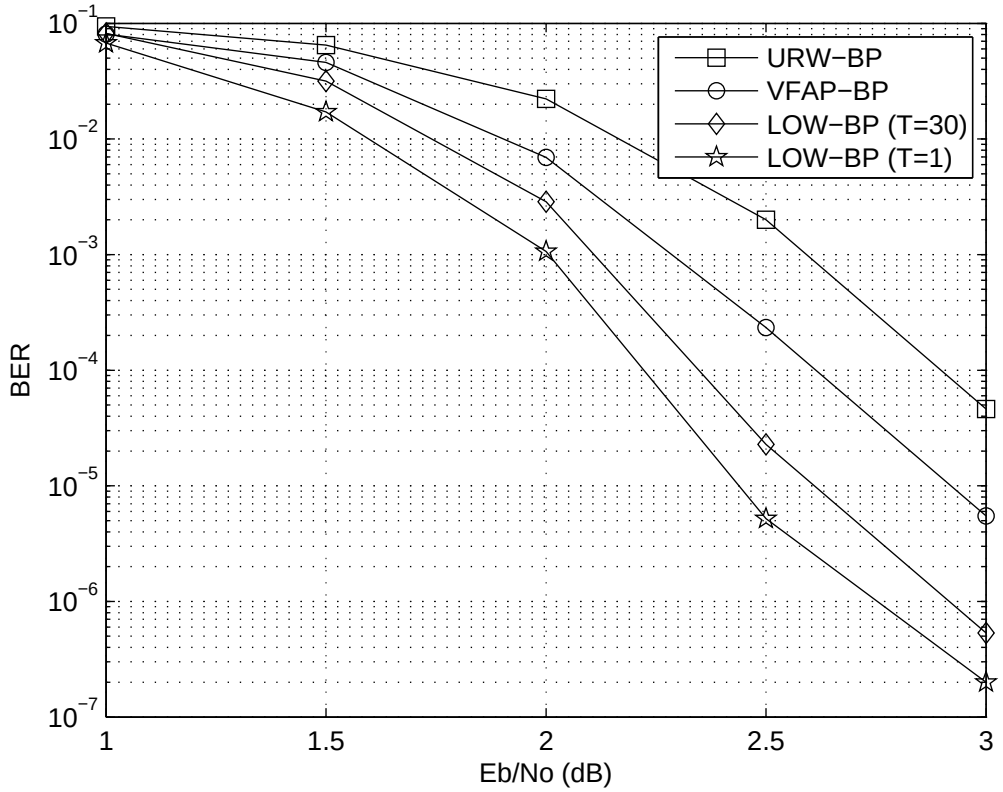

Figure 5.8: Comparison of the performance of BP (same results as URW-BP), URW-BP, VFAP-BP and the proposed LOW-BP for decoding irregular codes based on the min-sum decoding algorithm.

more effective to resolve the over-concentration effect, which complies with the argument in [84]: the min-sum rule rends to overshoot the LLRs more than the sum-product. Moreover, Fig. 5.8 demonstrates that the proposed LOW-BP algorithm applied with the min-sum decoding algorithm can further increase the performance gap to 0.7dB between the proposed and some of the existing decoding algorithms.

5.6 Summary

We have proposed a LOW-BP algorithm for decoding finite-length LDPC codes. The proposed algorithm has been compared to previously reported reweighted belief propagation algorithms and has demonstrated superior performance for the scenarios considered. LOW-BP comprises an offline and an online stage. The online stage relies on standard

tree-reweighted belief propagation, while the offline stage involves an optimization problem over subgraphs of the original factor graph. By increasing the number of subgraphs, the offline stage converges faster and exhibits less complexity. Reducing the number of subgraphs will lead to improved BER performance, albeit at an additional delay and complexity cost during the offline stage. LOW-BP is especially well suited to the decoding of short to moderate LDPC codes and is a promising choice for applications that require a reduced number of decoding iterations. Future avenues of research include fast adaptation of the offline stage to time-varying channel conditions. In the next chapter, we incorporate the proposed reweighted decoding algorithms in Chapter 4 and 5, i.e. VFAP-BP and LOW-BP algorithms, with distinct detectors to design iterative detection and decoding (IDD) receivers for multi-antenna systems.

Chapter 6

Iterative Detection and Decoding for Multi-Antenna Systems with Finite-Length LDPC Codes

Contents

6.1 Overview

In this chapter, we consider the problem of iterative detection and decoding (IDD) for multi-antenna systems using LDPC codes. Firstly, we present two novel QR decomposition (QRD)-based MIMO detectors which can approach the performance of the maximum *a posteriori* (MAP) algorithm with significant complexity reduction as compared

with the optimal MAP solution. The multiple-feedback QRD (MF-QRD) detector employs a multi-feedback technique to select appropriate candidates when the symbols are unreliable. Another detection strategy, called variable-M QRD (VM-QRD), is developed which dynamically adjusts the number of detection candidates according to the channel variations in each detection layer. Then, both proposed detectors are combined with the standard BP decoder in order to constitute QRD-type IDD receivers for MIMO systems. Finally, we present KA-type IDD receivers consisting of a soft PIC detector with linear MMSE receive filters and advanced LDPC decoders, as proposed in the previous chapters, i.e. the VFAP-BP and LOW-BP decoders. They are called the KA-type receivers since the proposed BP algorithms exploit the knowledge of short cycles in the graph structure and the reweighting factors derived from the hypergraph's expansion. Simulation results show that the computational complexity of QRD-type IDD receivers is largely reduced by using the proposed detectors while their performance can still approach the optimal solution with acceptable BER performance loss. On the other hand, when the proposed reweighted BP decoders are applied, the KA-type IDD receiver is able to outperform other receivers with existing decoding schemes and, more importantly, achieve low delay due to a small number of decoding iterations required.

6.2 Introduction

MIMO systems can support several independent data streams, resulting in a significant increase in system capacity [93]. In order to separate the data streams and mitigate the interference between them, a detection algorithm must be employed at the receiver end. In the last decade or so, a great deal of effort has been devoted to the development of detection algorithms and their integration with channel decoding techniques [18], [49], [94]- [105]. In this chapter, MIMO systems with joint detection/decoding are shown to produce excellent results, approaching the performance of an interference free scenario. In a system with joint detection/decoding an ideal receiver is comprised of two components: an efficient soft-input soft-output (SISO) MIMO signal detector and a SISO decoder with short delay. Specifically, the estimated log likelihood ratios associated with the encoded bits are computed by the detector and these estimates will serve as input to the decoder. Then, in the second phase of the detection/decoding iteration, the decoder generates *a posteriori*

probabilities for the encoded bits of each data stream. As a result, a soft estimate of the transmitted symbol is obtained which can facilitate detection in the first phase of the next outer iteration. The joint process of detection/decoding is then repeated in an iterative manner until the maximum number of iterations is reached. However, in practice, there are many open issues for such an IDD scheme, e.g. severe detection/decoding delays, especially for codes with short block lengths [49], [95], or prohibitively high computational complexity associated with IDD systems in general.

In a spatial multiplexing configuration, the system capacity increases linearly with the number of individual transmission streams from the transmitter to the receiver [106]. To separate all data streams with their respective spatial signatures, a variety of detection techniques have been studied, in which the optimal performance can be achieved at the expense of a very high complexity. The limitations of optimal detection schemes have motivated the exploration of sub-optimal detectors, such as those with zero-forcing (ZF) and MMSE receive filters, and ordered successive interference cancellation (O-SIC) or vertical BLAST (V-BLAST) schemes [107, 108], which can maintain low complexity but which may have poor performance. In [109], the authors develop a detector based on QRD which offers a comparable performance to O-SIC and further saves on computational effort through avoiding pseudo-inverse calculation at each detection layer. More recently, when applied with the lattice-reduction (LR) [110] or M-algorithm [111], the QRD detector has shown enhanced performance, even if additional computational cost is involved.

As for the decoding aspects of an IDD receiver, the standard BP algorithm is well known as the most effective algorithm to decode LDPC codes [7] and has been widely employed in IDD schemes for MIMO systems [49], [97] and [112]. Nonetheless, as mentioned in previous chapters, with the existence of cycles, the standard BP algorithm has a number of shortcomings, such as convergence to a codeword not being guaranteed and possibly taking many iterations, especially in low SNR regimes, which significantly deteriorates the decoding performance and causes unexpected transmission delays. Hence, many applications of LDPC-coded MIMO systems suffer performance degradation to some extent.

This chapter presents two types of IDD schemes for MIMO systems operating with a

spatial multiplexing configuration with reduced complexity and short delay. Inspired by other error propagation mitigation work [98,99], [100,104] and [101,113], QRD-type IDD receivers comprise a QRD-based detector, i.e. a MF-QRD or VM-QRD, and a normal BP decoder. In this scenario, both the MF-QRD and VM-QRD detectors are successfully incorporated with standard BP decoders. QRD-type receivers with an MF-QRD detection technique can achieve performance with a gap of 0.2dB in the first detector/decoder iteration in an LDPC-coded system with an optimal MAP detector. Equipped with a VM-QRD detector, QRD-type IDD receivers can also approach optimal performance as well as offer good gains after a few outer iterations. It should be noted that both proposed detectors introduce little extra complexity when compared to similar techniques. On the other hand, KA-type IDD receivers consist of a SISO PIC detector with linear MMSE receive filters and a more sophisticated decoder, i.e. VFAP-BP or LOW-BP which were elaborated in Chapters 4 and 5, respectively. Incorporated with a SISO PIC-MMSE detector, both KA reweighted decoding algorithms are shown to outperform existing schemes and considerably improve the performance of IDD processing in MIMO systems.

The organization of this chapter is as follows: Section 6.3 introduces the basic notation and the system model. In Section 6.4, the proposed QRD-type and KA-type IDD receivers are explained in detail. Section 6.5 shows the simulation results along with a discussion. Finally, Section 6.6 concludes the chapter.

6.3 System Model and Basic Notation

In this section, we briefly describe an uncoded MIMO spatial multiplexing system, an LDPC-coded MIMO spatial multiplexing system that performs iterative detection and decoding, and the basic notation used in subsequent sections.

6.3.1 Uncoded MIMO Spatial Multiplexing System Model

Now let us consider a spatial multiplexing MIMO system, as depicted in Fig. 6.1, with N_T transmit antennas and N_R receive antennas, where $N_R \geq N_T$. Given a time in-

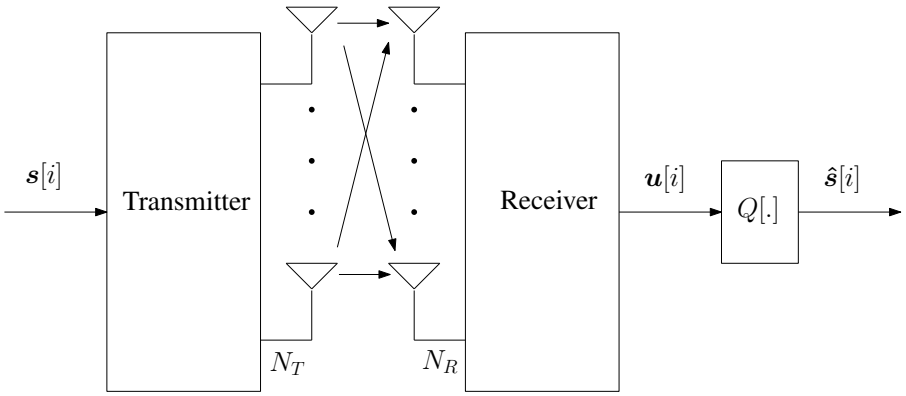

Figure 6.1: MIMO spatial multiplexing system.

dex $[i]$, the transmitter sends out N_T symbols which are denoted by an $N_T \times 1$ vector $\boldsymbol{s}[i] = [s_1[i], s_2[i], \ldots, s_{N_T}[i]]^T$ in which all the elements are taken from a constellation set $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$, where $(\cdot)^T$ denotes transpose and $|\mathcal{A}|$ denotes the total number of constellation points provided with a modulation scheme. The symbol vector $\boldsymbol{s}[i]$ is transmitted over a flat fading channel, then the signals are demodulated and sampled at the receiver end equipped with N_R antennas. The received signal is organized into an $N_R \times 1$ vector $\boldsymbol{y}[i] = [y_1[i], y_2[i], \ldots, y_{N_R}[i]]^T$ along with sufficient statistics for detection, as given by:

$$\boldsymbol{y}[i] = \sum_{k=1}^{N_T} \boldsymbol{c_k} s_k[i] + \boldsymbol{n}[i] = \boldsymbol{C}\boldsymbol{s}[i] + \boldsymbol{n}[i], \tag{6.1}$$

where the $N_R \times 1$ vector $\boldsymbol{n}[i]$ is zero mean complex circular symmetric Gaussian noise with covariance matrix $\mathbb{E}[\boldsymbol{n}[i]\boldsymbol{n}^H[i]] = \sigma_n^2 \boldsymbol{I}$, in which $\mathbb{E}[\cdot]$ represents the expected value, $(\cdot)^H$ denotes the Hermitian operator, σ_n^2 is the noise variance and $\boldsymbol{I}$ is the identity matrix. The symbol vector $\boldsymbol{s}[i]$ has a covariance matrix $\mathbb{E}[\boldsymbol{s}[i]\boldsymbol{s}^H[i]] = \sigma_s^2 \boldsymbol{I}$ where σ_s^2 is average signal power. The $N_R \times N_T$ channel matrix $\boldsymbol{C}$ consists of the complex element C_{n_R,n_T}, which corresponds to the channel impulse response from the n_T-th transmit antenna to the n_R-th receive antenna. The $Q[\cdot]$ in Fig. 6.1 stands for a quantization process to obtain the estimated signal $\hat{\boldsymbol{s}}[i]$ by slicing the soft decision $\boldsymbol{u}[i]$.

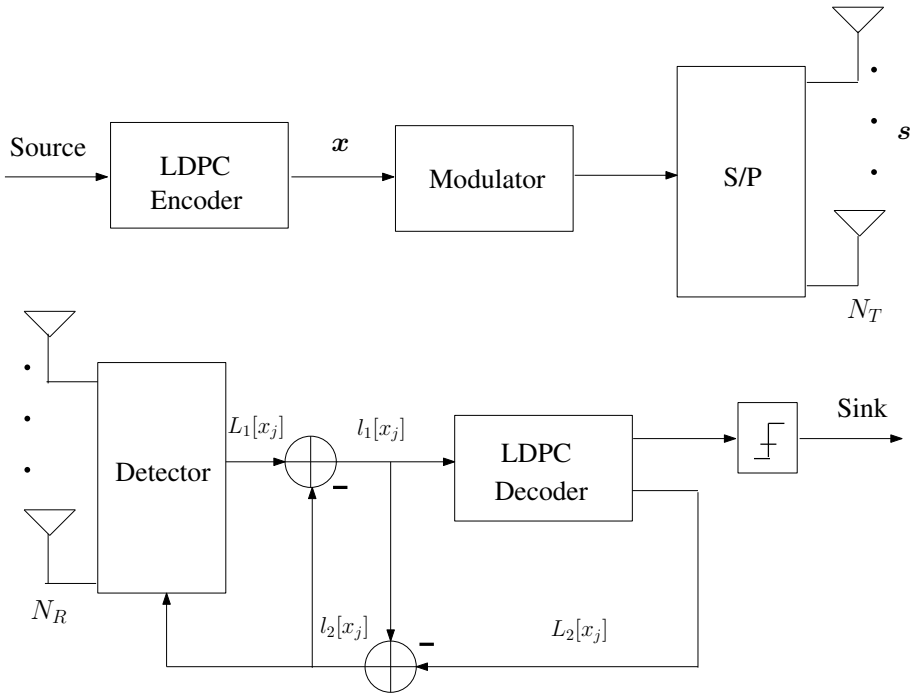

Figure 6.2: Iterative LDPC-coded MIMO spatial multiplexing system.

6.3.2 Iterative LDPC-Coded MIMO Spatial Multiplexing System

Dropping the term $[i]$ for simplicity, a block diagram of the IDD system employed in this work is depicted in Fig. 6.2, where the received data, after being demodulated, matched filtered and sampled, is collected in a vector $\boldsymbol{y} \in \mathbb{C}^{N_R \times 1}$ (here $\mathbb{C}$ denotes complex numbers) with sufficient statistics for detection. $\boldsymbol{s} \in \mathbb{C}^{N_T \times 1}$ is the transmitted symbol vector. In what follows, we assume that the receiver has perfect knowledge of the channel matrix $\boldsymbol{C} \in \mathbb{C}^{N_R \times N_T}$. Otherwise, an estimation algorithm must be employed to compute the entries in $\boldsymbol{C}$ [103, 105]. At the receiver end the received signal is detected and decoded iteratively by exchanging mutual soft information between the proposed detectors and the LDPC decoder. Throughout this chapter, we refer to the iterations between the detector and the decoder as outer iterations and the iterations within the LDPC decoder as inner iterations. Given any time instant, let x_j be the j-th bit of the constellation symbol vector ($j = 1, 2, \ldots, N_T \log_2 |\mathcal{A}|$) where $|\mathcal{A}|$ is the constellation size. Denote $L[x_j]$ as the LLR value for the LDPC coded bits x_j, which is defined as:

$$L[x_j] = \log P(x_j = 1) - \log P(x_j = 0). \tag{6.2}$$

The detector computes the *a posteriori* log-likelihood ratio (LLR) $L_1[x_j]$ of the j-th LDPC coded bit by using:

$$L_1[x_j] = \log P(x_j = 1|\boldsymbol{y}) - \log P\left(x_j = 0|\,\boldsymbol{y}\right), \tag{6.3}$$

which is based on the channel observation $\boldsymbol{y}$ and the *a priori* information $l_2^p[x_j]$ from the LDPC decoder. Let the superscript $(\cdot)^p$ denote the information obtained from the previous outer iterations. Using Bayes's rule, $L_1[x_j]$ can be rewritten as:

$$\begin{aligned} L_1[x_j] &= \log \frac{P(\boldsymbol{y}|x_j = 1)}{P(\boldsymbol{y}|x_j = 0)} + \log \frac{P(x_j = 1)}{P(x_j = 0)} \\ &= l_1[x_j] + l_2^p[x_j], \end{aligned} \tag{6.4}$$

where $l_2^p[x_j] = \log \frac{P(x_j=1)}{P(x_j=0)}$ is zero in the initial outer iteration. The *extrinsic* information $l_1[x_j]$ from the detector is computed by:

$$\begin{aligned} l_1[x_j] &= L_1[x_j] - l_2^p[x_j], \\ &= \log \frac{P(x_j = 1|\boldsymbol{y})}{P(x_j = 0|\boldsymbol{y})} - \log \frac{P(x_j = 1)}{P(x_j = 0)}, \\ &= \log \frac{\sum_{\boldsymbol{s}\in\mathcal{A}_j^+} P\big(\boldsymbol{y}|\boldsymbol{s}\big) \exp\big(L_a(\boldsymbol{s})\big)}{\sum_{\boldsymbol{s}\in\mathcal{A}_j^-} P\big(\boldsymbol{y}|\boldsymbol{s}\big) \exp\big(L_a(\boldsymbol{s})\big)}, \end{aligned} \tag{6.5}$$

where $\mathcal{A}_j^+$ is the set of all symbol vectors that consist of bits satisfying $x_j = 1$ and $\mathcal{A}_j^-$ is similarly defined as satisfying $x_j = 0$. The joint probability density $P\big(\boldsymbol{y}\big|\boldsymbol{s}\big)$ in (6.5) is derived in [18] as:

$$\mathrm{P}\big(\boldsymbol{y}\big|\boldsymbol{s}\big) \propto \exp\Big(-\frac{1}{\sigma_{eff}^2}\|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{s}\|^2\Big), \tag{6.6}$$

where σ_{eff}^2 is the variance of the effective noise-plus (remaining) interference (NPI) of $\hat{\boldsymbol{s}}$, and $L_a(\boldsymbol{s})$ denotes the *a priori* symbol probability for symbol vector $\boldsymbol{s}$. Notice that by using the Jacobian algorithm the computation of $l_1[x_j]$ can be simplified as:

$$l_1[x_j] \approx \frac{1}{2}\max_{\boldsymbol{s}\in\mathcal{A}_j^+}\Big\{-\frac{1}{\sigma^2}\|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{s}\|^2 + L_a(\boldsymbol{s})\Big\} - \frac{1}{2}\max_{\boldsymbol{s}\in\mathcal{A}_j^-}\Big\{-\frac{1}{\sigma^2}\|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{s}\|^2 + L_a(\boldsymbol{s})\Big\}. \tag{6.7}$$

It should be noted that for the reduced-complexity detectors proposed in Section 6.4, the probability densities of all possible transmitted vectors in (6.7) are not necessarily considered. Yet, this does not significantly affect the overall performance of the proposed detectors when compared to an optimal detector. The proposed detection algorithms carefully

select so-called tentative decisions in order to create an appropriate list $\mathcal{B}$. Moreover, we replace the sets $\mathcal{A}_j^+$ and $\mathcal{A}_j^-$ in (6.7) by $\mathcal{A}_{k,j}^+ \bigcap \mathcal{B}$ and $\mathcal{A}_{k,j}^- \bigcap \mathcal{B}$ respectively, as follows:

$$\begin{aligned} l_1[x_j] \approx & \frac{1}{2} \max_{\boldsymbol{s} \in \mathcal{A}_{k,j}^+ \bigcap \mathcal{B}} \Big\{ -\frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{s}\|^2 + L_a(\boldsymbol{s}) \Big\} \\ & - \frac{1}{2} \max_{\boldsymbol{s} \in \mathcal{A}_{k,j}^- \bigcap \mathcal{B}} \Big\{ -\frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{s}\|^2 + L_a(\boldsymbol{s}) \Big\}. \end{aligned} \tag{6.8}$$

In the subsequent section, more details on how to generate the list $\mathcal{B}$ for the proposed detection schemes are presented. In the second part of the outer iteration, the LDPC decoder outputs the *a posteriori* LLR with respect to the j-th coded bit, $L_2[x_j]$, by computing:

$$\begin{aligned} L_2[x_j] &= \log \frac{P[x_j = 1 | l_1^p[x_j; \text{decoding}]]}{P[x_j = 0 | l_1^p[x_j; \text{decoding}]]} \\ &= l_2[x_j] + l_1^p[x_j]. \end{aligned} \tag{6.9}$$

Notice that the reliability of this *a posteriori* information mainly depends on the decoding algorithm employed as well as the maximum number of inner iterations. After subtracting $l_1^p[x_j]$ the remainder of $L_2[x_j]$, denoted by $l_2[x_j]$, will be fed back to the detector as *a priori* information in order to update $L_1[x_j]$ for the next outer iteration. The above IDD resumes until the maximum number of outer iterations is reached. Finally, the estimated symbol vector $\hat{\boldsymbol{s}}$ is determined by a hard decision.

6.4 Proposed IDD Receivers for MIMO Systems

In this section, we propose two types of IDD receiver for spatial multiplexing MIMO systems. QRD-type receivers consist of the proposed MF-QRD or VM-QRD detectors and a standard BP decoder. Both detectors can approach optimal performance with much less computational complexity than the MAP algorithm. The KA-type receivers are formed by a soft PIC detector and KA reweighted BP decoders which were presented in Chapters 4 and 5. By applying advanced BP decoding, KA-type receivers improve the performance of the IDD process significantly at no additional computational cost during reception.

6.4.1 Proposed QRD-type IDD Receivers

As for the QRD-type receivers, the detection part in Fig. 6.2 is performed by either of the proposed detectors, i.e. the MF-QRD or VM-QRD detector, while a standard BP decoder is empowered in the decoding part. Since the standard BP or sum-product decoding algorithm was elaborated in Chapter 2, this section only focuses on describing the proposed detection algorithms. Both MF-QRD and VM-QRD detection schemes are developed on the basis of QRD detection [109]. As an alternative to the BLAST algorithms, the QRD algorithm transforms the channel matrix $\boldsymbol{C}$ as:

$$\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{R}, \tag{6.10}$$

where $\boldsymbol{Q}$ is an $N_R \times N_T$ matrix having orthogonal columns with unit norm and $\boldsymbol{R}$ is an $N_T \times N_T$ upper triangular matrix obtained by QR decomposition. By combining the received signal $\boldsymbol{y}[i]$, now the sufficient statistics for detecting $\boldsymbol{s}[i]$ become:

$$\tilde{\boldsymbol{y}}[i] = \boldsymbol{Q}^{\boldsymbol{H}}\boldsymbol{y}[i] = \boldsymbol{R}\boldsymbol{s}[i] + \boldsymbol{\eta}[i], \tag{6.11}$$

where the noise vector $\boldsymbol{\eta}[i] = \boldsymbol{Q}^{\boldsymbol{H}}\boldsymbol{n}[\boldsymbol{i}]$ has the same statistical characteristics. For the purpose of detecting the signal element $\tilde{s}_k[i]$ in the k-th layer ($k = 1, 2, \ldots, N_T$) the corresponding soft decision $u_k[i]$ is computed as:

$$u_k[i] = \frac{\tilde{\boldsymbol{y}}[i] - \sum_{\tau=k+1}^{N_T} r_{k,\tau}\hat{s}_\tau[i]}{r_{k,k}}, \tag{6.12}$$

where $r_{k,k}$ are the diagonal entries of $\boldsymbol{R}$. It is noted in this chapter that the detection ordering is determined by the signal-to-interference-plus-noise ratio (SINR), such that signals with higher post-detection SINR will be detected first. The post-detection SINR is achieved by calculating the linear MMSE with the equation below:

$$\text{SINR}_k = \frac{\sigma_s^2 |\boldsymbol{\omega}_{k,\text{MMSE}}\boldsymbol{c}_k|^2}{\sigma_s^2 \sum_{l \neq k} |\boldsymbol{\omega}_{k,\text{MMSE}}\boldsymbol{c}_l|^2 + \sigma_n^2 \|\boldsymbol{\omega}_{k,\text{MMSE}}\|^2}, \tag{6.13}$$

where $\boldsymbol{c}_k$ is the k-th column vector of the channel matrix $\boldsymbol{C}$ and $\boldsymbol{\omega}_{k,\text{MMSE}}$ is the k-th row of the MMSE matrix defined as:

$$\boldsymbol{\Omega}_{\text{MMSE}} = (\boldsymbol{C}^H\boldsymbol{C} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{C}^H. \tag{6.14}$$

Throughout this chapter, the above criterion is applied in order to determine the optimal detection order.

Table 6.1: The Pseudo Code of the MF-QRD Algorithm

Initialization:

$\boldsymbol{QR} = \boldsymbol{C}$;

$\tilde{\boldsymbol{y}} = \boldsymbol{Q}^H \boldsymbol{y}$;

Multiple Feedback QRD Detection:

for $k = N_T, \ldots, 1$

$u_k[i] = (\tilde{y} - \sum_{\iota=k+1}^{N_T} r_{k,\iota} \hat{s}_\iota)/r_{k,k}$;

if $u_k[i]$ **is unreliable**

$\boldsymbol{\mathcal{L}} = [c_1, c_2, \ldots, c_m, \ldots, c_M]^T$;

$\tilde{y}'_k = \tilde{y} - \sum_{\iota=k+1}^{N_T} r_{k,\iota} \hat{s}_\iota$;

for $m = 1$ **to** M **do**

for $q = k-1$ **to** 1 **do**

$\hat{b}_q^m = \left(\tilde{y}'_k - r_{k,k} c_m - \sum_{\tau=q+1}^{N_T} r_{q,\tau} \hat{b}_\tau^m\right)/r_{q,q}$;

end for

$\boldsymbol{b}_k^m[i] = \left[\hat{b}_1^m, \ldots, \hat{b}_q^m, \ldots, c_m, \hat{\boldsymbol{s}}_{k+1,N_T}\right]$;

end for

$m_{\text{opt}} = \underset{1 \le m \le M}{\operatorname{argmin}} ||\boldsymbol{y}[i] - \boldsymbol{C}\boldsymbol{b}_k^m[i]||^2$;

$\hat{s}_k[i] = c_{\text{opt}}$;

else

$\hat{s}_k[i] = Q\left[u_k[i]\right]$;

end if

end for

Proposed MF-QRD Detection Algorithm

As a low-complexity interference cancellation strategy, MF processing has been well exploited to achieve feedback gains in some detection scenarios, such as MF-SIC [104] and MF-IDD [105]. In this section, we explore the idea of integrating the MF processing with a conventional QRD detector. Our proposed MF-QRD detection scheme is described in Table 6.1 in which the received signals are recovered from the N_T-th layer to the first layer. At any k-th layer, a soft decision $u_k[i]$ is calculated by using (6.12), then a decision feedback procedure is carried out. At the beginning of the decision feedback, a threshold d_{th} is defined according to [114] which states that d_{th} can be an empirical value based on experiments or be a function of channel characteristics. In this thesis, d_{th} is set to 0.1 for

all scenarios. For each k-th layer we compute d_k, the norm of the difference between $u_k[i]$ and its nearest constellation points. The estimate $u_k[i]$ is considered reliable if $d_k < d_{th}$ otherwise $u_k[i]$ is determined as unreliable. In the former case, we obtain $\hat{s}_k[i]$ by using $\hat{s}_k[i] = Q\big[u_k[i]\big]$. In the latter case where $u_k[i]$ is unreliable, we generate a candidate vector $\boldsymbol{\mathcal{L}} = [c_1, c_2, \ldots, c_m, \ldots, c_M]^T$ that is a selection of the closest M constellation points to $u_k[i]$ and the value of M can be either fixed or flexible which results in distinct complexities. Based on various c_m at the k-th layer, the estimated signal $\hat{b}_q^m$ from the $k-1$-th layer to the first layer can be computed as:

$$\hat{b}_q^m = \big(\tilde{y}_k' - r_{k,k}c_m - \sum_{\tau=q+1}^{N_T} r_{q,\tau}\hat{b}_\tau^m\big)/r_{q,q}, \qquad q = k-1, \ldots, 1, \tag{6.15}$$

Thus, for each c_m we have the following row vector:

$$\boldsymbol{b}_k^m[i] = \big[\hat{b}_1^m, \ldots, \hat{b}_q^m, \ldots, c_m, \hat{\boldsymbol{s}}_{k+1,N_T}\big], \tag{6.16}$$

where the row vector $\hat{\boldsymbol{s}}_{k+1,N_T} = [\hat{s}_{k+1}, \hat{s}_{k+2}, \ldots, \hat{s}_{N_T}]$ stands for the existing detected signals from the $k+1$-th layer to the N_T-th layer. At last, the optimum index m_{opt} is chosen under an ML criterion which is:

$$m_{\text{opt}} = \underset{1 \le m \le M}{\operatorname{argmin}} ||\boldsymbol{y}[i] - \boldsymbol{C}\boldsymbol{b}_k^m[i]||^2, \tag{6.17}$$

such that the corresponding c_{opt} is determined as $\hat{s}_k[i]$.

Proposed VM-QRD detection

The second proposed detection scheme is the VM-QRD algorithm, described in Table 6.2, which is capable of approaching ML performance while saving significantly on computational cost. The reduction in complexity is achieved by improving the existing channel-based intra radius selection (CBIRS) [115] with a new norm differential screening (NDS) strategy. The QRM-MLD algorithm proceeds by selecting a group of M candidates at every detection layer. Near-optimal performance can be achieved when M is fixed and equal to the constellation size, which also results in relatively high complexity. On the

Table 6.2: The Pseudo Code of the VM-QRD Algorithm

Initialization:

$\boldsymbol{QR} = \boldsymbol{C}$;

$\tilde{\boldsymbol{y}} = \boldsymbol{Q}^H \boldsymbol{y}$;

Variable-M QRD:

$f_k(x) = \sum_{\gamma=1}^{k} \|\tilde{y} - \sum_{\tau=\gamma+1}^{N_T} r_{\gamma,\tau} x_\tau\|^2$;

Step 1

$\{\tilde{s}_{4,m,1}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_1(x)$

Step 2

if $P_{m,k} < d_{th}^2$ **and** reliable in (6.20)

$\{\tilde{s}_{3,m,2}, \tilde{s}_{4,m,1}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{4,m,1}\}_{m=1})$

else

$\{\tilde{s}_{3,m,2}, \tilde{s}_{4,m,1}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{4,m,1}\}_{m=1}^{M})$

end

Step 3

if $P_{m,k} < d_{th}^3$ **and** reliable in (6.20)

$\{\tilde{s}_{2,m,3}, \tilde{s}_{3,m,3}\tilde{s}_{4,m,3}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{3,m,2}, \tilde{s}_{4,m,1}\}_{m=1})$

else

$\{\tilde{s}_{2,m,3}, \tilde{s}_{3,m,3}\tilde{s}_{4,m,3}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{3,m,2}, \tilde{s}_{4,m,1}\}_{m=1}^{M})$

end

. .

. .

Step N_T

if $P_{m,k} < d_{th}^{N_T}$ **and** reliable in (6.20)

$\{\tilde{s}_{1,m,N_T}, \ldots \tilde{s}_{N_T,m,N_T}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{2,m,N_T-1}, \ldots, \tilde{s}_{N_T,m,N_T-1}\}_{m=1})$

else

$\{\tilde{s}_{1,m,N_T}, \ldots \tilde{s}_{N_T,m,N_T}\}_{m=1}^{M} = \underset{M}{\operatorname{argmin}}\, f_2(x, \{\tilde{s}_{2,m,N_T-1}, \ldots, \tilde{s}_{N_T,m,N_T-1}\}_{m=1}^{M})$

end

other hand, a large performance degradation takes place if the value of M is too small. The proposed VM-QRD scheme employs a CBIRS as well as an NDS strategy to enable the value of M to be flexible, depending on the CSI in each layer. In the following, the CBIRS algorithm is first explained which defines a threshold C_k $(k = N_T, \ldots, 1)$ for adjusting M. Then, we discuss how the NDS strategy affects the final candidate(s).

Channel-Based Intra Radius Selection

We consider the threshold C_k $(k = N_T, \ldots, 1)$ to constrain the size of M. The parameter C_k is updated according to the probability ε of discarding the ML partial weight $P_{ml,k}$ at the k-th layer. The relation between C_k and $P_{ml,k}$ is given by:

$$\Pr\{P_{ml,k} > P_{b,k} + C_k | P_{ml,k} > P_{b,k}\} < \varepsilon, \tag{6.18}$$

where $P_{(b,k)} = \|\tilde{\boldsymbol{y}}_k - \boldsymbol{R}_k \hat{\boldsymbol{s}}_{b,k}\|^2$. By denoting $\boldsymbol{R}_k$ as the last k columns and rows in $\boldsymbol{R}$, and letting $\tilde{\boldsymbol{y}}_k$ and $\hat{\boldsymbol{s}}_{b,k}$ represent the last k entries of both vectors, the threshold C_k can also be derived in a complex value form as:

$$C_k = 4\sigma_v^2 \|\boldsymbol{r}_w\| \mathcal{Q}^{-1}(\varepsilon \mathcal{Q}(\frac{\|\boldsymbol{r}_w\|}{2\sigma_v^2})) - \|\boldsymbol{r}_w\|^2 \tag{6.19}$$

where $\mathcal{Q}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-x^2}{2}} dx$, and $\boldsymbol{r}_w$ denotes the "weakest" column of $\boldsymbol{R}_k$ (the column with the smallest norm). Once $d_{th}^k = P_{(b,k)} + C_k$ is specified, due to this restriction we have the following selection rule: if the partial weight $P_{m,k}$ is larger than d_{th}^k, then the corresponding candidate is eliminated; otherwise it will be saved as a remaining candidate.

Norm Differential Screening Strategy

The NDS strategy is applied to filter the remaining candidates selected further by CBIRS. Note that all remaining candidates' partial weights are sorted in an increasing order. Hence, by computing the difference between the first two survival candidates' partial weights, we can determine whether the first candidate is accurate enough for the current detection layer.

$$\begin{cases} P_{1,k} - P_{2,k} \leq \rho, \ M = M_s, \ \text{(unreliable)} \\ P_{1,k} - P_{2,k} > \rho, \ M = 1, \ \text{(reliable)} \end{cases} \tag{6.20}$$

where M_s is the number of candidates selected by CBIRS, $P_{1,k}$ and $P_{2,k}$ denote the first two sorted partial weights. At this point, the first candidate corresponding to $P_{1,k}$ is chosen as $\hat{s}_k[i]$ directly if it is considered as reliable; otherwise all the candidates selected by CBIRS are stored for the $K+1$-th detection layer.

6.4.2 Proposed KA-type IDD Receivers

The proposed KA-type receivers are composed of a SISO PIC detector with MMSE receive filters and one of the KA reweighted BP decoders proposed in Chapters 4 and 5, respectively. In previous chapters, VFAP-BP and LOW-BP decoders were shown to offer outstanding performance as well as low decoding latency, i.e. fewer decoding iterations. More importantly, both decoding algorithms optimise the reweighted parameters $\boldsymbol{\rho}$ offline which indicates that no extra online complexity is needed. Thus, we expect that the combination of a simple detection scheme and the reweighted decoding algorithms can enhance the performance of IDD receivers without requiring much computational cost. In the sequel, we depict the proposed KA-type IDD receivers by splitting them into two procedures, namely, the computation of the soft symbols using PIC detection and that of the *a posteriori* LLRs using KA decoding.

Calculation of *a posteriori* LLRs Using Parallel Interference Cancellation (PIC) Detection

In a SISO PIC-MMSE detection algorithm, estimates of the transmitted symbols are obtained based on the *a priori* LLRs obtained from the LDPC channel decoder. These "soft" estimates are extracted from the received vector to perform interference cancellation for a MIMO system. The remaining NPI terms are then equalized by a linear MMSE receive filter which is followed by computation of the a posteriori LLRs of the individual constituent bits. The SISO PIC-MMSE algorithm used as an outer component is detailed in the following.

Based on the SISO model in [94], a PIC detector processes the kth data stream and

cancels the interference from all other streams ($q \neq k$) such that:

$$\hat{\boldsymbol{y}}_k = \boldsymbol{y} - \sum_{q \neq k} \boldsymbol{c}_q \hat{s}_q = \boldsymbol{c}_k s_k + \tilde{\boldsymbol{n}}, \quad \forall k \tag{6.21}$$

where $\hat{s}_q, q \neq k$ are soft estimates of the transmitted co-channel symbols obtained from the channel decoder which are computed according to $\hat{s}_q = \mathbb{E}[s_q] = \sum_{a_c \in \mathcal{A}} P[y_q = a_c] \times a_c$ where $P[s_q = a_c]$ corresponds to the *a priori* probability of the symbol $a_c \in \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ on the constellation map $\mathcal{A}$. In (6.21), the term $\boldsymbol{c}_k$ is the kth column of the channel matrix $\boldsymbol{C}$ and $\tilde{\boldsymbol{n}}$ is the remaining NPI vector corresponding to:

$$\tilde{\boldsymbol{n}} = \sum_{q \neq k} \boldsymbol{c}_q e_q + \boldsymbol{n}, \tag{6.22}$$

in which $\boldsymbol{n}$ is the actual channel noise in (6.1) and e_q denotes the error between the transmitted symbol s_q and the soft symbol $\hat{s_q}$ as:

$$e_q = s_q - \hat{s_q}. \tag{6.23}$$

For the next step, the NPI vector $\tilde{\boldsymbol{n}}$ is to be equalized by a linear MMSE receive filters as follows:

$$\hat{s}_k = \tilde{\boldsymbol{w}}_k^H \hat{\boldsymbol{y}}_k = \tilde{\boldsymbol{w}}_k^H \boldsymbol{c}_k s_k + \tilde{\boldsymbol{w}}_k^H \tilde{\boldsymbol{n}}, \tag{6.24}$$

where '$(\cdot)^H$' denotes the Hermitian transpose and the MMSE receive filter is derived by using:

$$\tilde{\boldsymbol{w}}_k^H = E_s \boldsymbol{c}_k^H \big(\boldsymbol{C} \tilde{\boldsymbol{\Lambda}}_k \boldsymbol{C}^H + N_0 \boldsymbol{I}_{N_R} \big), \tag{6.25}$$

with the real-valued $N_T \times N_T$ diagonal matrix $\tilde{\boldsymbol{\Lambda}}_k$ consisting of entries:

$$\tilde{\Lambda}_{q,q} = \begin{cases} E_q, q \neq k \\ E_s, q = k \end{cases} \tag{6.26}$$

where E_s represents the transmission energy while E_q denotes the variance of the estimation error as $E_q = \mathrm{Var}[s_q] = \mathbb{E}[|e_q|^2]$. Note that (6.25) requires an $N_R \times N_R$-dimensional matrix inversion to be carried out for each stream, each received vector and each outer iteration. Subsequently, in the k-th layer, the coded bit extrinsic LLR of the j-th bit can be calculated by using:

$$l_1[x_{k,j}] = \log \frac{\sum_{a_c \in \mathcal{A}_j^+} P(\hat{s_k} | s_k = a_c) \prod_{(i \neq j)} P(x_{k,i})}{\sum_{a_c \in \mathcal{A}_j^-} P(\hat{s_k} | s_k = a_c) \prod_{(i \neq j)} P(x_{k,i})} \tag{6.27}$$

where $\mathrm{P}\big(\hat{s_k} \big| s_k = a_c\big)$ is derived by:

$$\mathrm{P}\big(\hat{s_k} \big| s_k = a_c\big) = \frac{1}{\pi \sigma_{eff}^2} \exp\Big(-\frac{1}{\sigma_{eff}^2} \|\hat{s_k} - s_k\|^2 \Big), \tag{6.28}$$

such that (6.27) is equivalent to:

$$l_1[x_{k,j}] = \log \frac{\sum_{a_c \in \mathcal{A}_j^+} \exp(-|\hat{s_k} - s_k|^2/\sigma_{\text{eff}}^2) \prod_{(i \neq j)} P(x_{k,i})}{\sum_{a_c \in \mathcal{A}_j^-} \exp(-|\hat{s_k} - s_k|^2/\sigma_{\text{eff}}^2) \prod_{(i \neq j)} P(x_{k,i})}, \tag{6.29}$$

with $P(x_{k,i})$ being the *a priori* probability of $x_{k,i}$ the i-th bit in the k-th stream as:

$$P(x_{k,i}) = \frac{1}{2}[1 + x_{k,i} \tanh(\frac{1}{2} l_2^p[x_{k,i}])], \tag{6.30}$$

which is based on the its soft symbol $l_2^p[x_{k,i}]$ derived from a previous outer iteration.

Computation of Soft Symbols using KA Reweighted Decoding algorithms

The extrinsic information $l_1[\boldsymbol{x}]$, obtained from the SISO PIC detector, is then fed to the LDPC decoder as *a priori* information. According to (6.9), the KA reweighted LDPC decoder outputs the *a posteriori* LLR with respect to the j-th coded bit, i.e. $L_2[x_j]$ which is equivalent to $\lambda_{\text{Belief},j}$ used in Chapter 5. Designed to improve the convergence behaviour of the standard BP algorithm, KA reweighted decoding algorithms refer to the VFAP-BP and LOW-BP algorithms proposed in Chapters 4 and 5, respectively. In Tables 6.4.2 and 6.4.2, we present the algorithm flows of KA-type IDD receivers which employ either VFAP-BP or LOW-BP decoding algorithms. More details regarding their offline and online operations can be found in each dedicated chapter. At the end of decoding, $\lambda_{\text{Belief},j}$ serves as soft output for deciding the value of $\hat{x}_j$ or for generating the extrinsic information $l_2[x_j]$ in the next IDD iteration.

6.5 Simulation Results

In this section, numerical results are presented with respect to testing both QRD-type and KA-type IDD receivers. As for the proposed QRD-type receiver, we start by examining the two proposed detection algorithms in an uncoded MIMO system, then incorporate them with a standard BP decoder to form an IDD structure. We consider two channel models, i.e. a 4×4 MIMO system with uncorrelated Rayleigh flat fading and the 3GPP spatial channel model (SCM) [116] developed for assessing MIMO systems in an outdoor environment. The 3GPP SCM is operating at a centre frequency of 2 GHz with a system

Table 6.3: Algorithm Flow of KA-type IDD Receivers Using a VFAP-BP Decoder

Offline stage: determination of ρ for the hypergraph

1: Run the algorithms [15, 16] to count the number of cycles with girth length passing each check node;

2: Determine variable FAPs for each check node given two reweighting parameters, i.e. 1 and $2/\bar{n_D}$;

Online Stage: real-time detection and decoding

3: For all data streams, use (6.27)-(6.30) to calculate extrinsic information $l_1[\boldsymbol{x}]$ and send it to VFAP-BP decoder;

4: With the *a priori* information from the SISO PIC detector, update the soft symbol outputs $L_2[\boldsymbol{x}]$ iteratively using the reweighted message passing rules (4.5)-(4.7) until the maximum number of inner decoding iterations is reached;

5: If the maximum number of outer iterations is reached, output $\hat{\boldsymbol{x}}$ by hard decision; otherwise subtract $l_1^p[\boldsymbol{x}]$ from $L_2[\boldsymbol{x}]$ then feed $l_2[\boldsymbol{x}]$ back to the SISO PIC detector.

Table 6.4: Algorithm Flow of KA-type IDD Receivers Using a LOW-BP Decoder

Offline stage: determination of ρ for the hypergraph

1: Apply the modified PEG algorithm to expand the original hypergraph into $T \geq 1$ subgraphs;

2: For the tth subgraph initialize $\boldsymbol{\rho}_t$ to a valid value, then compute the beliefs as well as the mutual information terms by use the reweighted message passing rules (5.13)-(5.15);

3: Update $\boldsymbol{\rho}_t^{(r)}$ to $\boldsymbol{\rho}_t^{(r+1)}$ using the conditional gradient method;

4: Repeat steps 2-3 until $\boldsymbol{\rho}_t$ converges, then determine the final $\boldsymbol{\rho}$ for all check nodes;

Online Stage: real-time detection and decoding

5: For all data streams, use (6.27)-(6.30) to calculate extrinsic information $l_1[\boldsymbol{x}]$ and send it to the LOW-BP decoder;

6: With the *a priori* information from the SISO PIC detector, update the soft symbol outputs $L_2[\boldsymbol{x}]$ iteratively using reweighted message passing rules (5.13)-(5.15) until the maximum number of inner decoding iterations is reached;

7: If the maximum number of outer iterations is reached, output $\hat{\boldsymbol{x}}$ by a hard decision; otherwise subtract $l_1^p[\boldsymbol{x}]$ from $L_2[\boldsymbol{x}]$ then feed $l_2[\boldsymbol{x}]$ back to the SISO PIC detector.

bandwidth of 5 MHz. For QPSK modulation, $M = 4$ and $d_{th} = 0.5$ for MF-QRD, the complex lattice reduction [117] is implemented for LR-QRD and about 50% of detection

complexity is reduced compared to the real lattice reduction algorithm. Four constellation candidates are preserved at each layer for the QRM-MLD scheme. For 16-QAM modulation, we have $M = 8$ and $d_{th} = 0.5$ for MF-QRD, and 16 constellation candidates are preserved at each layer for QRM-MLD scheme.

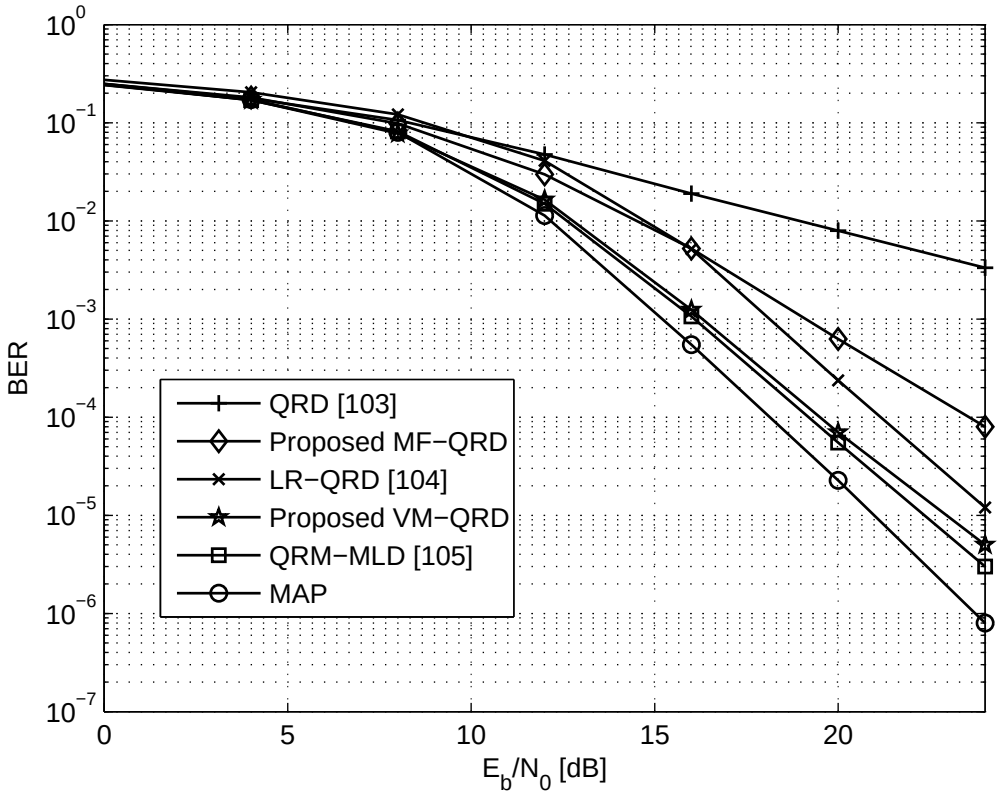

Figure 6.3: BER performance comparisons of the proposed QRD-based detectors with a conventional QRD detector [109], an LR-QRD detector [110] and a QRM-MLD detector [111] using a flat fading channel with 16-QAM modulation.

Firstly, we compare the proposed QRD-based uncoded detection schemes with a conventional QRD detector [109], an LR-QRD detector [110] and a QRM-MLD detector [111] in terms of BER performance as well as computational complexity. As shown in Fig. 6.3, both proposed detectors are capable of significantly improving the conventional QRD detector [109] and approaching the optimal performance represented by a MAP detector. Moreover, notice that within the low/moderate SNR range (under 15 dB) the proposed MF-QRD detector can even outperform an LR-QRD detector [110]. An analysis of the computational complexities of QRD-based detectors is presented in Fig. 6.4, where

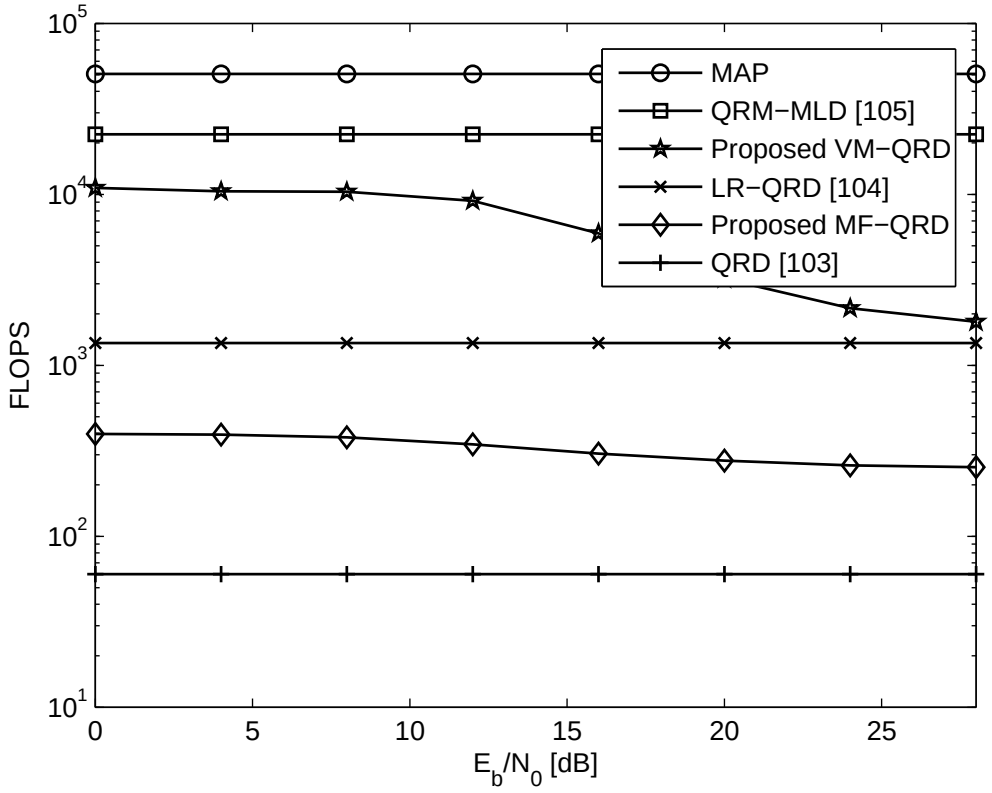

Figure 6.4: Computational complexity in FLOPS of the proposed QRD-based detectors, the conventional QRD detector [109], the LR-QRD detector [110] and the QRM-MLD detector [111] using flat fading channel with 16-QAM modulation. Note that the complexity of proposed VM-QRD algorithm reduces when SNR increases, which is because the first detection candidate corresponding to $P_{1,k}$ is always considered reliable in the high SNR regime.

the average detection complexity is evaluated by the average number of floating-point operations (FLOPS) required. This part of the simulation is executed by the Lightspeed toolbox [118] in which the number of FLOPS equals 2 for a complex addition and 6 for a multiplication. Based on results from Fig. 6.3 and Fig. 6.4, it is obvious that the proposed MF-QRD and VM-QRD detection schemes are able to achieve near-optimal BER performance at relatively low computational costs when compared to other existing suboptimal QRD-based detection algorithms. Next, in Fig. 6.5, we further test the performance of the proposed detection schemes in more practical 3GPP SCM with MIMO spatial multiplexing ($N_T = N_R = 4$). Similar to the previous results, both proposed MF-QRD and

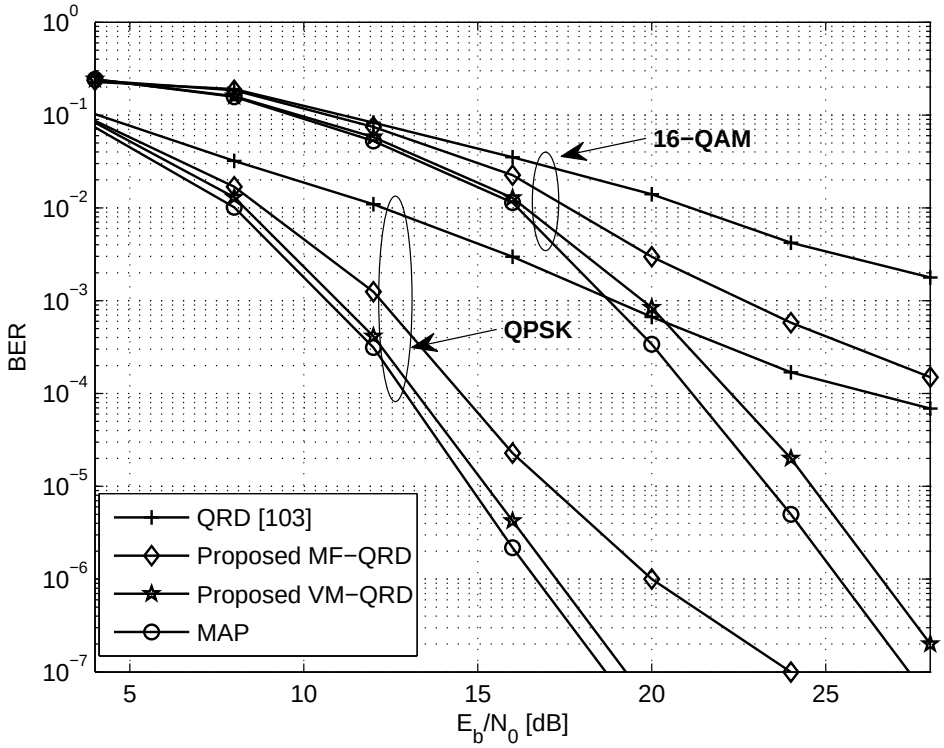

Figure 6.5: BER performance comparisons of the proposed QRD-based detectors with a conventional QRD detector [109] and a MAP detector using 3GPP SCM with QPSK modulation and 16-QAM modulation.

VM-QRD detectors can enhance a conventional QRD detector [109] significantly and approach the optimal performance generated by a MAP detector.

Finally, we evaluate the performance of the proposed QRD-type IDD receivers which are developed based on the proposed detection algorithms and a standard BP decoding algorithm. The irregular LDPC code used is generated by the PEG algorithm [21] with variable nodes' distribution $\upsilon(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$ and check nodes' distribution $\nu(x) = x^5$, block length $N = 1000$, code rate $R = 0.5$ and a maximum of 50 decoding iterations. In Fig. 6.6, in the first outer iteration the MF-QRD detection with the multi-branch (MB) processing algorithm [113] contributes to the QRD-type receiver approaching optimum performance by nearly 0.2 dB, but only requiring similar computational complexity as conventional QRD detection. Additionally, a QRD-type receiver with the proposed VM-QRD detection algorithm can even outperform the one with MF-QRD detection algorithm by a small margin. After 3 outer iterations, the QRD-type receiver with VM-QRD detection algorithm is able to provide another 0.5 dB of gain while the QRD-type receiver with an MF-QRD detection algorithm does not

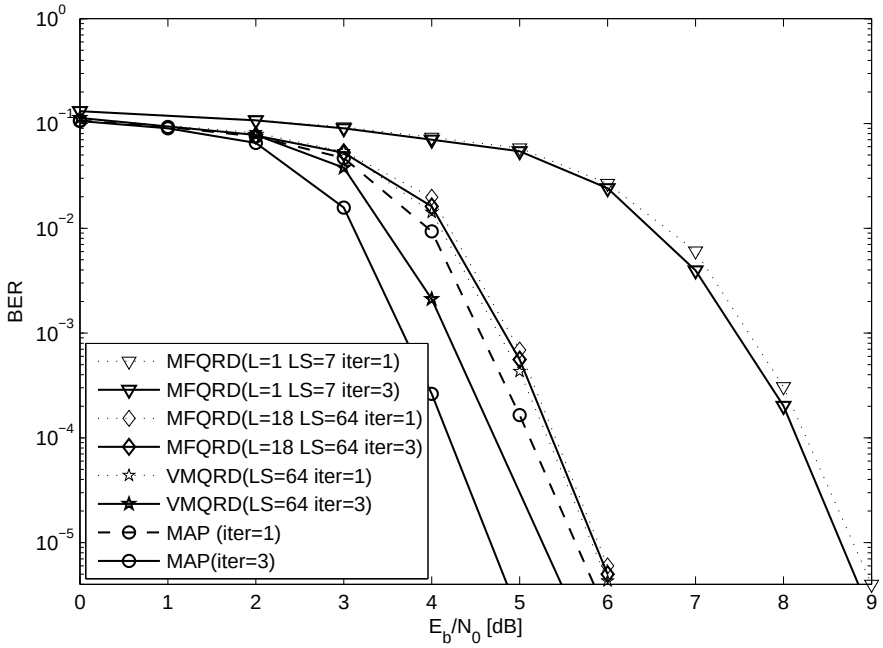

Figure 6.6: BER performance comparisons of LDPC-coded QRD-based detectors with an LDPC-coded MAP detector using a flat fading channel with QPSK modulation, where 'L' denotes branch size, 'LS' denotes the list size and 'iter' is the number of outer iterations

offer much performance gain from more outer iterations. It should be noted that the performance of the proposed MF-QRD detection algorithm significantly deteriorates in the case where only a single branch is available.

Now we present the simulation results of the proposed KA-type IDD receiver consisting of the reweighted decoding algorithms proposed in previous chapters and a PIC-MMSE detection algorithm for a 4×4 MIMO system with QPSK modulation. The LDPC code used here is a regular code designed by the PEG algorithm [21] with block length $N = 1000$, the rate $R = 0.5$, the girth g is 6, and the degree distributions are $3(\upsilon(x) = x^4)$ and $5(\nu(x) = x^6)$, respectively. To illustrate the performance with a constraint on the decoding latency, the inner decoding iterations are reduced to 30. For the LOW-BP decoder, $T = 20$ subgraphs are generated, where check nodes are allowed to be revisited, and 600 recursions are employed to obtain $\boldsymbol{\rho}$. In comparison with the standard BP and URW-BP algorithms, we first draw EXIT chart of the different decoders with a SISO PIC detector in Fig. 6.7. Although the curve of the PIC-MMSE detector does not reach the top-right $(1, 1)$ point at the given SNR, it is obvious that the combination of a PIC-MMSE detector and the proposed LOW-BP decoder creates the largest detection and decoding tunnel.

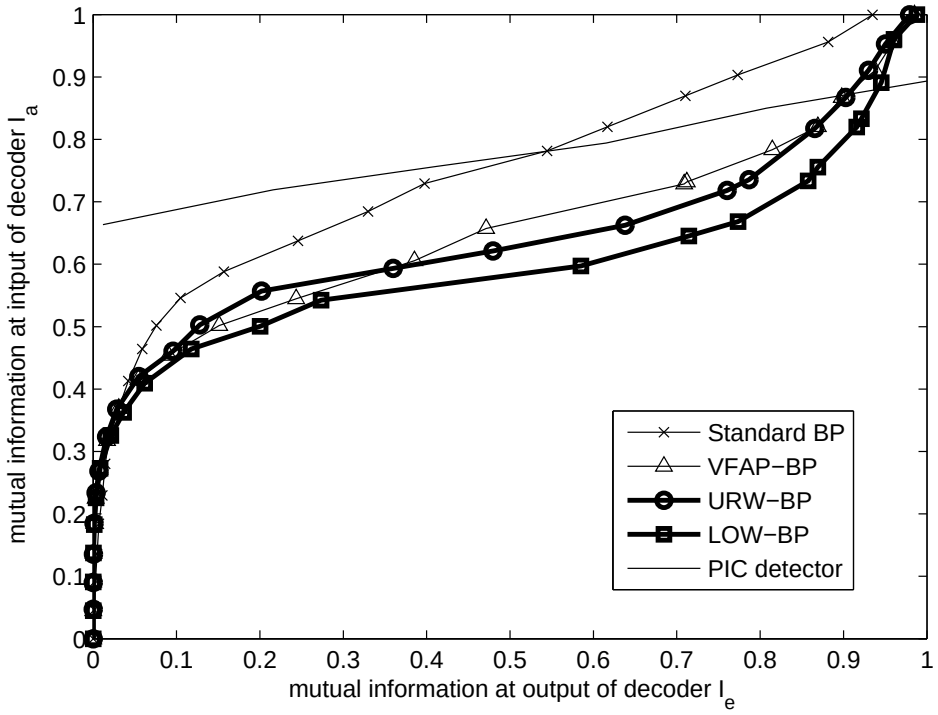

Figure 6.7: EXIT charts for different decoders and the PIC detector. The LOW-BP decoder matches better with the PIC detector than other decoders. The EXIT chart of the PIC detector is obtained at $E_b/N_0 = 4$dB.

Additionally, only the tunnel between the PIC-MMSE detector and the standard BP decoder is closed at an early stage, which indicates that the performance gain from the IDD process could be significantly diminished in this case.

To verify the result of the EXIT chart, Fig. 6.8 depicts the BER performance of the proposed KA-type receiver. We use 30 inner decoding iterations and 3 outer iterations. The proposed KA-type receiver with either a VFAP-BP or LOW-BP decoding algorithm can outperform a receiver with a standard BP or URW-BP algorithm in the first outer iteration. In the third outer iteration, the proposed KA-type receiver is still able to generate relatively good performance when considering the low SNR range and the block length of code. Notice that there is an error floor effect at a BER of 10^6, which can be mitigated by using decision feedback techniques, [98], [104] and [105]. As mentioned above, the key feature of the proposed KA-BP decoders lies in that no additional complexity is imposed in real-time decoding since the optimization of $\boldsymbol{\rho}$ is carried out offline. Moreover, by increasing the number of subgraphs T LOW-BP can accelerate the optimization process,

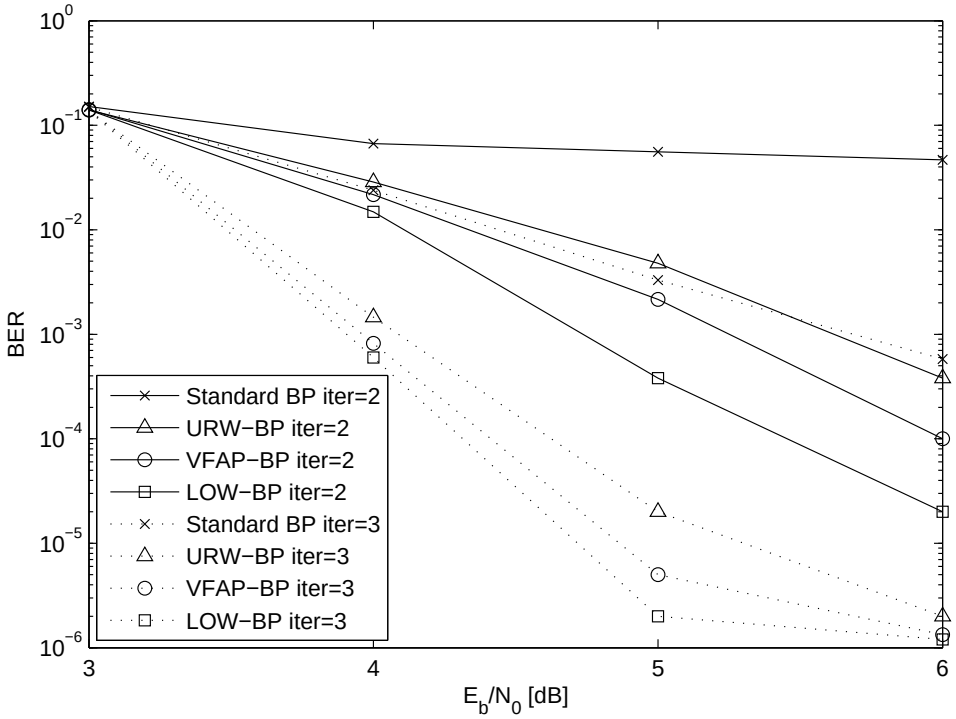

Figure 6.8: Performance comparison of the KA-type IDD receiver with standard BP, URW-BP, VFAP-BP and LOW-BP for a 4×4 MIMO system using QPSK modulation. 'iter' stands for the number of outer iterations.

which enables KA-type IDD receivers to be employed for time-varying channels.

6.6 Summary

In this chapter two types of IDD receivers, based on LDPC-coded MIMO systems, have been proposed which utilise a combination of novel QRD detection schemes and standard BP decoding algorithms or a combination of a basic detection scheme and advanced decoding algorithms. For the QRD-type receiver, the proposed MF-QRD and VM-QRD detection schemes are shown to produce good performance with tradeoffs between BER performance and computational complexity by adjusting parameters, i.e. M and d_{th}. Equipped with either detection schemes, the QRD-type IDD receiver can offer near-optimal performance while not requiring many outer iterations. Facilitated by the reweighted decoding algorithms proposed in earlier chapters, the KA-type IDD receivers

can outperform other receivers with both alternative reweighted and the standard BP algorithms. Another advantage of KA-type receivers is that even if the VFAP-BP and the LOW-BP algorithms have different computational costs in the optimization phase, neither of them requires extra complexity for online decoding.

Chapter 7

Conclusions and Future Work

Contents

7.1 Summary of Work

In this thesis, effective encoding and decoding schemes have been investigated with respect to implementing finite-length LDPC codes in wireless communication systems. We have focused on the exploitation of the graphical structure of the TG of the codes and the use of a number of mathematical tools, such as algorithms for counting short cycles [15, 16], the ACE metric [14, 19] and the conditional gradient method, in order to reduce the performance degradation of LDPC codes due to the presence of short cycles. Based on the rich literature of LDPC codes, various techniques and algorithms have been developed corresponding to research topics of practical importance, i.e. RC code design, reweighted decoding strategies and IDD receivers, which can be efficiently employed in modern communication systems. In the following, we summarise the work reported in each chapter of this thesis.

In Chapter 3, three puncturing and two extension schemes are proposed for designing regular and irregular RC LDPC codes with short/moderate block lengths. By applying

algorithms for counting cycles, the ACE spectrum and an exhaustive search, the proposed RC techniques are able to achieve a wide range of rates $(0.1 < R < 0.9)$ and largely minimise the performance loss caused by puncturing or extension. To achieve low rates $(R < 0.5)$, the proposed extension techniques are preferred as they can be developed on a layered-structured model with linear-time encoding. On the other hand, to generate RC codes at high rates $(R > 0.5)$,the proposed puncturing algorithms have been shown to have an advantage in terms of overall performance. In addition, ACE-based puncturing can generate puncturing patterns for irregular LDPC codes that provide comparable performance to a SIM-based puncturing method, which illustrates the effectiveness of the proposed RC designs.

In Chapter 4, a novel reweighted BP decoding algorithm is presented which mitigates the effect of short cycles by reweighting part of the extrinsic message exchanged between nodes in the TG. The proposed VFAP-BP algorithm has been shown to improve the convergence behavior of the standard BP decoding algorithm within a limited number of iterations, which is desirable in wireless communications as low delay or latency is often required. Moreover, compared to existing reweighted algorithms, the VFAP-BP algorithm is applicable to decoding both regular and irregular LDPC codes.

Chapter 5 extended the line of reweighted strategies to optimise the reweighting parameters locally offline, i.e. independent of real-time transmissions. Unlike VFAP-BP or other existing reweighted decoding algorithms, e.g. URW-BP [20], the reweighting parameters acquired from the proposed LOW-BP algorithm are not constrained to one or two values. Furthermore, when compared to the TRW-BP algorithm [17], the LOW-BP decoding algorithm offers a trade-off between overall performance and the convergence speed of the reweighting parameters by spliting the factor graph into different numbers of subgraphs and subsequently performing local optimization. During the offline stage, the convergence of reweighting parameters can be accelerated by increasing the number of subgraphs. On the other hand, reducing the number of subgraphs will enhance the decoding performance at the expense of slow convergence and extra complexity. As mentioned earlier, similar to the VFAP-BP algorithm, the LOW-BP algorithm does not add any redundant computational complexity to the online decoding phase.

Finally, Chapter 6 has introduced two IDD receivers for the spatial multiplexing of

MIMO systems. The QRD-type receiver combines either the proposed MF-QRD or VM-QRD detection algorithm with a standard BP or sum-product decoding algorithm, while the KA-type receiver is facilitated by simple SISO PIC detection and one of the reweighted decoding techniques proposed in the previous chapters. In comparison with other detection algorithms, the proposed MF-QRD and VM-QRD schemes have been verified to approach the optimal performance generated by a MAP detector but only requires a realistic computational complexity. Our experimental results have shown that a QRD-type IDD receiver can provide near-optimal performance without many outer iterations, and a KA-type IDD receiver can outperform receivers equipped with other decoding schemes.

7.2 Future Work

First of all, the framework and algorithms reported in this thesis are applicable to various channel models with more complex transmission scenarios [7, 9]. Secondly, we have mainly focused on finite-length LDPC codes constructed by random computer-based design methods. Other design techniques, such as combinatorial [6] or geometry-based methods [7], are also available to explore. Finally, all proposed encoding and decoding algorithms can, potentially, be extended from binary LDPC codes to non-binary LDPC codes.

Future avenues of relevant research are listed as follows:

- In terms of the design of RC LDPC codes, the proposed puncturing and extension schemes can be employed in more structured RC models, e.g. [71], which enables efficient hardware implementations. Moreover, the CC-based and ACE-based techniques are applicable to other approaches for generating RC LDPC codes, e.g. masking [7] or information shortening [78].
- The combination of short-cycle counting with the "divide and concur" strategy [90] can potentially improve the VFAP-BP algorithm, especially promising for application to complexity-reduced decoders, e.g. min-sum and max-product decoders [92].

- For the LOW-BP decoding algorithm, future work includes fast adaptation of the offline stage to time-varying channel conditions and a detailed analysis regarding the convergence behaviour of reweighting parameters in the optimisation stage.
- As for Chapter 6, the application of QRD-type and KA-type IDD receivers can be extended from point-to-point scenarios to multiple-user, multiple-cell and massive MIMO scenarios.
- It is possible to devise combined strategies of advanced estimation algorithms [119]- [135] with proposed decoding techniques for LDPC-coded systems.

Glossary

ACE	**A**pproximate **C**ycle **E**MD
ARQ	**A**utomatic **R**epeat-**Q**equest
AWGN	**A**dditive **W**hite-**G**aussian-**N**oise
BEC	**B**inary **E**rasure **C**hannel
BER	**B**it-**E**rror **R**ate
BI-AWGN	**B**inary-**I**nput **A**dditive **W**hite-**G**aussian-**N**oise
BP	**B**elief **P**ropagation
BSC	**B**inary **S**ymmetric **C**hannel
CBIRS	**C**hannel-**B**ased **I**ntra **R**adius **S**election
CC-based	**C**ycle-**C**ounting based
CSI	**C**hannel **S**tate **I**nformation
DE	**D**ensity **E**volution
EAP	**E**dge **A**ppearance **P**robability
EMD	**E**xtrinsic **M**essage **D**egree
EXIT	**EX**trinsic-**I**nformation-**T**ransfer
FAP	**F**actor **A**ppearance **P**robability
FEC	**F**orward **E**rror **C**orrection
FER	**F**rame-**E**rror **R**ate
FLOPS	**FL**oating-point **OP**eration**S**
GE	**G**aussian **E**limination
IDD	**I**terative **D**etection and **D**ecoding
i.i.d.	**i**ndependently **i**dentically **d**istributed
KA-BP	**K**nowledge-**A**ided **B**elief **P**ropagation
LDPC	**L**ow-**D**ensity **P**arity-**C**heck
LLR	**L**og-**L**iklihood **R**atio
LOW-BP	**L**ocally-**O**ptimized re**W**eighted **B**elief **P**ropagation
LP	**L**inear **P**rogramming
LR	**L**attice-**R**eduction
MAP	**M**aximum ***A Posteriori***
MB	**M**ulti-**B**ranch

MF-QRD	**M**ultiple-**F**eedback **QR D**ecomposition
MIMO	**M**ultiple-**I**nput **M**ultiple-**O**utput
ML	**M**aximum-**L**ikelihood
MMSE	**M**inimum **M**ean **S**quare **E**rror
NDS	**N**orm **D**ifferential **S**creening
NPI	**N**oise-**P**lus (remaining) **I**nterference
O-SIC	**O**rdered **S**uccessive **I**nterference **C**ancelation
pdf	**p**robability **d**ensity **f**unction
PEG	**P**rogressive **E**dge-**G**rowth
PIC	**P**arallel **I**nterference **C**ancelation
QC-LDPC	**Q**uasi-**C**yclic **L**ow-**D**ensity **P**arity-**C**heck
QRD	**QR D**ecomposition
RA	**R**e-**A**ppearance
RC	**R**ate-**C**ompatible
SCM	**S**patial **C**hannel **M**odel
SIC	**S**uccessive **I**nterference **C**ancelation
SIM-based	**SIM**ulation-based
SINR	**S**ignal-to-**I**nterference-plus-**N**oise **R**atio
SISO	**S**oft-**I**nput **S**oft-**O**utput
SNR	**S**ignal-to-**N**oise-**R**atio
SPA	**S**um-**P**roduct **A**lgorithm
TG	**T**anner **G**raph
TRW-BP	**T**ree-**R**e**W**eighted **B**elief **P**ropagation
URW-BP	**U**niformly **R**e**W**eighted **B**elief **P**ropagation
V-BLAST	**V**ertical-**B**ell **L**aboratories L**A**yered **S**pace-**T**ime
VFAP-BP	**V**ariable **F**actor **A**ppearance **P**robabilities **B**elief **P**ropagation
VM-QRD	**V**ariable-**M QR D**ecomposition
WER	**W**ord-**E**rror **R**ate
ZF	**Z**ero-**F**orcing

Bibliography

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, pp. 379-423, 623-656, July, October, 1948.

[2] D. J. C. Mackay, "Information Theory, Inference, and Learning Algorithms," Cambridge University Press, Version 6.6 , December, 2003.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proceedings IEEE International Conference on Communications,* Geneva, Switzerland, pp. 1064-1070, May 1993.

[4] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457-458, March 1997.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399-431, March 1999.

[6] T. Richardson and R. Urbanke, "Modern Coding Theory," Cambridge University Press, May 2008.

[7] W. Ryan and S. Lin, "Channel Codes: Classical and Modern," Cambridge University Press, 1st edition, Oct. 30, 2009.

[8] A. Burr, "Modulation and Coding for Wireless Communications," Prentice Hall, 1st edition, June 20, 2001.

[9] T. K. Moon, "Error Correction Coding: Mathematical Methods and Algorithms," Wiley-Blackwell, July 1, 2005.

[10] R. G. Gallager, “Low-density parity check codes,” *IRE Trans. Inf. Theory.*, vol. 39, no. 1, pp. 37-45, Jan. 1962.

[11] Y. Kou, S. Lin, M. P. C. Fossorier, “Low-density parity-check codes based on finite geometries: a rediscovery and new results,” *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711-2736, Nov. 2001.

[12] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sep. 1981.

[13] H. Wymeersch, “Iterative Receiver Design, “Cambridge University Press, Sep. 2007.

[14] Tian Tao, C. R. Jones, J. D. Villasenor, R. D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Transaction on Communications*, vol. 52, no. 8, pp. 1242- 1247, Aug. 2004.

[15] T. R. Halford, K. M. Chugg, “An algorithm for counting short cycles in bipartite graphs,” *IEEE Trans. Information Theory*, vol. 52, no. 1, pp. 287-292, Jan. 2006.

[16] M. Karimi, A. H. Banihashemi, “A message-passing algorithm for counting short cycles in a graph,” *2010 IEEE Information Theory Workshop (ITW)*, vol., no., pp. 1-5, 6-8 Jan. 2010.

[17] M. J. Wainwright, T. S. Jaakkola, and A.S. Willsky, “A new class of upper bounds on the log partition function,” *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2313-2335, July 2005.

[18] B. Hochwald and S. T. Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Transactions on Communications*, vol. 51, pp. 389-399, Mar. 2003.

[19] D. Vukobravic, V. Senk, “Evaluation and design of irregular LDPC codes using ACE spectrum,” *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2272-2279, Aug. 2009.

[20] H. Wymeersch, F. Penna and V. Savic, “Uniformly Reweighted Belief Propagation for Estimation and Detection in Wireless Networks,” *IEEE Trans. Wireless Communications*, vol. PP, No. 99, pp. 1-9, Feb. 2012.

[21] Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graph," *IEEE Transaction on Information Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

[22] C. T. Healy and R. C. de Lamare, "Decoder-Optimised Progressive Edge Growth Algorithms for the Design of LDPC Codes with Low Error Floors," *IEEE Communications Letters*, 2012.

[23] X. Wang and H. V. Poor, "Iterative (Turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 1046-1061, 1999.

[24] T. M. Cover and J. A. Thomas, "Elements of Information Theory," 2nd edition. Wiley-Interscience, 2006.

[25] R. G. Gallager, "Information Theory and Reliable Communication," Wiley, 2nd edition, 1968.

[26] S. Wilson, "Digital Modulation and Coding," Prentice-Hall, 1996.

[27] J. Proakis, "Digital Communications," McGraw-Hill, 4th edition, 2000.

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

[29] M. R. Yazdani, S. Hemati, A. H. Banihashemi, "Improving belief propagation on graphs with cycles," *IEEE Communications Letters*, vol. 8, no. 1, pp. 57- 59, Jan. 2004

[30] R. Chen, H. Huang, G Xiao, "Relation Between Parity-Check Matrixes and Cycles of Associated Tanner Graphs," *IEEE Communications Letters*, vol. 11, no. 8, pp. 674-676, August 2007.

[31] M. Sipser, D. A. Spielman, "Expander codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710-1722, Nov. 1996.

[32] N. Wiberg, "Codes and Decoding on General Graphs," Ph.D. disertation, Linköping University, Linköping, Sweeden, 1996.

[33] N. Wiberg, H. A. Loeliger, and R. Kötter, "Codes and Iterative Decoding on General Graphs," *Proc. IEEE Symp. Information Theory*, Adelaide, Australia, Septempber, 1995.

[34] Hui Jin, R. J. McEliece, "General coding theorems for turbo-like codes," *Proceedings IEEE International Symposium on Information Theory*, vol., no., pp. 120, 2000.

[35] Yongyi Mao, A. H. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," *IEEE International Conference on Communications*, vol. 1, no., pp. 41-44 Jun. 2001.

[36] S. Benedetto, G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591-600, May 1996.

[37] K. Chugg, A. Anastasopoulos, X. Chen, Iterative Detection: Adaptivity, Complexity Reduction, and Applications. Norwell, MA: Kluwer, 2001.

[38] N. Alon, R. Yuster and U. Zwick, "Finding and Counting Given Length Cycles," *Algorithmica*, Vol. 17 no. 3, pp. 209-223, 1997.

[39] Jun Fan, Yang Xiao, "A Method of Counting the Number of Cycles in LDPC Codes," *8th International Conference on Signal Processing*, vol. 3, no., pp. 16-20, 2006.

[40] R. Lucas, M. P. C. Fossorier, Yu Kou, Shu Lin, "Iterative decoding of one-step majority logic deductible codes based on belief propagation," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 931-937, June 2000.

[41] J. Xu, L. Chen, I. Djurdjevic, S. Lin, K. Abdel-Ghaffar, "Construction of Regular and Irregular LDPC Codes: Geometry Decomposition and Masking," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 121-134, Jan. 2007.

[42] D. J. C. MacKay, M. C. Davey, "Evaluation of Gallager Codes for Short Block Length and High Rate Applications," *Codes, Systems, and Graphical Models: Volume 123 of IMA Volumes in Mathematics and its Applications*, pp. 113-130, Spring-Verlag, New York, 2000.

[43] B. Vasic, "Structured iteratively decodable codes based on Steiner systems and their application in magnetic recording," *IEEE Global Telecommunications Conference*, vol. 5, no., pp. 2954-2960 vol. 5, 2001.

[44] B. Vasic, "Combinatorial constructions of low-density parity check codes for iterative decoding," *Proceedings IEEE International Symposium on Information Theory*, vol., no., pp. 312, 2002.

[45] S. J. Johnson, S. R. Weller, "Construction of low-density parity-check codes from Kirkman triple systems," *IEEE Global Telecommunications Conference*, vol. 2, no., pp. 970-974, vol. 2, 2001.

[46] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585-598, Feb. 2001.

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

[48] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727-1737, Oct. 2001.

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

[50] A. Ashikhmin, G. Kramer, S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657- 2673, Nov. 2004.

[51] S. ten Brink, G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2764-2772, Nov. 2003.

[52] Sae-Young Chung, G. D. Forney, T. J. Richardson, R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, Feb. 2001.

[53] M. Chiani, A. Ventura, "Design and performance evaluation of some high-rate irregular low-density parity-check codes," *IEEE Global Telecommunications Conference*, vol. 2, no., pp. 990-994 vol.2, 2001.

[54] M. Yang, W. E. Ryan, Yan Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Transactions on Communications*, vol.52, no.4, pp. 564- 571, April 2004.

[55] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Res. Labs, Cambridge, MA, Tech. Rep. TR2001-22, 2002.

[56] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Information Theory*, vol. 49, no. 5, pp. 1120-1146, May 2003.

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

[58] H. Xiao and A. H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Communications*, vol. 12, no. 12, pp. 498-519, Dec. 2004.

[59] R. G. Gallager, Low Density Parity Check Codes. MIT Press, 1963.

[60] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, X. Hu, "Reduced-Complexity Decoding of LDPC Codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005.

[61] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389-400, Apr. 1988.

[62] S. Lin and P. S. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Trans. Commun.*, vol. 30, no. 7, pp. 1701-1719, Jul. 1982.

[63] D.N. Rowitch, L.B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," , *IEEE Transactions on Communications*, vol. 48, no. 6, pp .948-959, Jun. 2000.

[64] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with different evolution," in *Proc. 2000 IEEE International Symposium on Information Theory (ISIT)*, pp. 5, Jun. 2000.

[65] C. Di, D. Proietti, I. E. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Infor. Theory*, vol. 48, pp. 1570-1579, Jun. 2002.

[66] G. Yue, B. Lu and X. Wang, "Analysis and design of finite-length LDPC codes," *IEEE Trans. Vehicular Technology.*, vol. 56, no. 3, May 2007.

[67] J. Li and K. R. Narayanan, "Rate-Compatible Low Density Parity Check (RC-LDPC) Codes for Capacity-Approaching ARQ Schemes in Packet Data Communications," *Proceeding of International Conference on on Communications, Internet and Information Technology (CIIT)*, US Virgin Islands, pp. 201-206, Nov. 2002.

[68] Jeongseok Ha, Jaehong Kim, S.W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Infor. Theory*, vol. 50, no. 11, pp. 2824- 2836, Nov. 2004.

[69] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short bolck length," *IEEE Trans. Infor. Theory*, vol. 52, no. 2, pp. 728-738, Feb. 2006.

[70] H. Y. Park, J. W. Kang, K. S. Kim, and K. C. Whang, "Efficient puncturing method for rate-compatible low-density parity-check codes," *IEEE TWC*, pp. 3914-3919, Nov. 2007.

[71] Jaehong Kim, A. Ramamoorthy, S. McLaughlin, "The design of efficiently-encodable rate-compatible LDPC codes," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 365-375, February 2009.

[72] M. El-khamy, J. Hou, and N. Bhushan, "Design of rate-compatible structured LDPC codes for hybrid ARQ applications," *IEEE JSAC*, pp. 965-973, Aug. 2009.

[73] H. Saeedi, H. Pishro-Nik, and A. H. Banihashemi, "Successive maximization for the systematic design of universally capacity approaching rate-compatible sequences of LDPC code ensembles over binary-input output-symmetric memoriless channels," *IEEE TCOM*, pp. 1807 - 1819, July 2011.

[74] H. Pishro-Nik and F. Fekri, "Result on punctured low-density parity-check codes and improved iterative decoding techniques," in *Proc. 2004 IEEE Infor. Theory Workshop*, pp. 24-29, San Antonio, TX, Oct. 2004.

[75] Hossein Pishro-Nik, Faramarz Fekri, "Results on Punctured Low-Density Parity-Check Codes and Improved Iterative Decoding Techniques," *IEEE Transaction on Information Theory*, vol. 53, no. 2, pp. 599-614, Feb. 2007.

[76] B. N. Vellambi and F. Fekri, "Finite-length rate-compatible LDPC codes: a novel puncturing scheme," *IEEE Transaction on Communications*, vol. 57, no. 2, pp. 297-301, Feb. 2009.

[77] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159-161, Mar. 2004.

[78] T. Tian, C. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP Journal on Wireless Communications and Networking.*, Vol. 2005, no. 5, Oct. 2005.

[79] Jingjing Liu, R. C. de Lamare, "Novel intentional puncturing schemes for finite-length irregular LDPC codes," *17th International Conference on Digital Signal Processing (DSP)*, vol., no., pp. 1-6, 6-8 July 2011.

[80] Jingjing Liu, R. C. de Lamare, "Finite-length rate-compatible LDPC codes based on extension techniques," *8th International Symposium on Wireless Communication Systems (ISWCS)*, vol., no., pp. 41-45, 6-9 Nov. 2011.

[81] Hua Xiao, A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Communications Letters*, vol. 8, no. 12, pp. 715- 717, Dec. 2004.

[82] T. J. Richardson, "Error Floor of LDPC Codes," in *Proc. 41st Annu. Allerton Conf. on Communication, Control, and Computing*, Urbana-Champaign, Oct. 2003, pp. 1426-1435.

[83] R. Mantha, F. R. Kschischang, "A capacity-approaching hybrid ARQ scheme using turbo codes," *Global Telecommunications Conference, 1999. GLOBECOM '99,* vol. 5, no., pp. 2341- 2345 vol. 5, 1999.

[84] H. Wymeersch, F. Penna and V. Savic, "Uniformly reweighted belief propagation: A factor graph approach," *2011 IEEE International Symposium on Information Theory Proceedings (ISIT)*, Issue Date: July 31 2011-Aug. 5 2011 On page(s): 2000 - 2004.

[85] IEEE 802.16e. Air interface for fixed and mobile broadband wireless access systems. IEEE P802.16e/D12 Draft, Oct 2005.

[86] T. G. Roosta, M. J. Wainwright and S. S. Sastry, "Convergence Analysis of Reweighted Sum-Product Algorithms," *IEEE Transactions on Signal Processing*, vol.56, no.9, pp.4293-4305, Sept. 2008.

[87] J. Feldman, M. Wainwright, and D. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954–972, 2005.

[88] S. Barman, X. Liu, S. C. Draper, and B. Recht, "Decomposition methods for large scale LP decoding," *preprint arXiv:1204.0556*.

[89] J. Liu, R. C. de Lamare, "Low-Latency Reweighted Belief Propagation Decoding for LDPC Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1660-1663, October 2012.

[90] J. S. Yedidia, Wang Yige, S. C. Draper, "Divide and Concur and Difference-Map BP Decoders for LDPC Codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 786-802, Feb. 2011.

[91] A. Uchoa, C. Healy, R. C. de Lamare and R. D. Souza, "Design of LDPC Codes Based on Progressive Edge Growth Techniques for Block Fading Channels," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1221-1223, 2012.

[92] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE International Conference on Communications*, vol. 2, pp. 1009–1013, 1995.

[93] I. E. Telatar, "Capacity of Multi-Antenna Gaussian Channels", *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585-595, 1999.

[94] X. Wang and H. V. Poor, "Iterative (Turbo) soft interferece cancellation and decoding for coded CDMA," *IEEE Trans. Communications*, vol. 47, No. 7, pp. 1046-1061, 1999.

[95] J. Hou, P. H. Siegel, L. B. Milstein, "Design of multi-input multi-output systems based on low-density Parity-check codes," *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 601- 611, April 2005.

[96] H. Lee, B. Lee, and I. Lee, "Iterative detection and decoding with an improved V-BLAST for MIMO-OFDM Systems, *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 504-513, Mar. 2006.

[97] Jianming Wu, Heung-No Lee, "Performance Analysis for LDPC-Coded Modulation in MIMO Multiple-Access Systems," *IEEE Transactions on Communications*, vol. 55, no. 7, pp. 1417-1426, July 2007.

[98] R.C. de Lamare, R. Sampaio-Neto, "Minimum Mean-Squared Error Iterative Successive Parallel Arbitrated Decision Feedback Detectors for DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 56, no. 5, pp. 778-789, May 2008.

[99] J. H. Choi, H. Y. Yu, Y. H. Lee, "adaptive MIMO decision feedback equalization for receivers with time-varying channels," *IEEE Trans. Signal Proc.*, 2005, 53, no. 11, pp. 4295-4303.

[100] R.C. de Lamare, R. Sampaio-Neto, "Adaptive MBER decision feedback multiuser receivers in frequency selective fading channels," *IEEE Communications Letters*, vol. 7, no. 2, Feb. 2003, pp. 73 - 75.

[101] R.C. de Lamare, R. Sampaio-Neto, A. Hjorungnes, "Joint iterative interference cancellation and parameter estimation for cdma systems," *IEEE Communications Letters*, vol. 11, no. 12, December 2007, pp. 916 - 918.

[102] J. W. Choi, A. C. Singer, J Lee, N. I. Cho, "improved linear soft-input soft-output detection via soft feedback successive interference cancellation," *IEEE Trans. Commun.*, vol. 58, no. 3, March 2010.

[103] R. C. de Lamare and R. Sampaio-Neto, "Adaptive reduced-rank equalization algorithms based on alternating optimization design techniques for MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 248-2494, Jul. 2011.

[104] P. Li, R. C. de Lamare and R. Fa, "Multiple Feedback Successive Interference Cancellation Detection for Multiuser MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2434-2439, August 2011.

[105] Peng Li, R. C. de Lamare, "Adaptive Decision-Feedback Detection With Constellation Constraints for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 853-859, Feb. 2012.

[106] P. W. Wolniansky, G. J. Foschini, G. D. Golden and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel", *URSI Int. Symposium on SSS.*, pp. 295, Oct 1998.

[107] G. D. Golden, C. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture", *Electronics Letters*, vol. 35, No. 1, January 1999.

[108] A. Rontogiannis, V. Kekatos, and K. Berberidis," A Square-Root Adaptive V-BLAST Algorithm for Fast Time-Varying MIMO Channels," *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 265-268, May 2006.

[109] R. Bohnke, D. Wubben, V. Kuhn and K.-D. Kammeyer, "Reduced complexity MMSE detection for BLAST architectures", *IEEE GLOBECOM '03.*, Vol. 4, pp. 2258, Dec. 2003.

[110] D. Wubben, R. Bohnke, V. Kuhn and K.-D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems", *ITG Workshop on Smart Antennas.*, pp. 106, Mar. 2004.

[111] K. J. Kim and J. Yue, "Joint channel estimation and data detection algorithms for MIMO-OFDM systems", *Proc. 36th Asilomar Conf. Signals System Comput.*, vol. 2, pp. 1857-1861, Nov. 2002.

[112] F. Ding, T. Chen, "Gradient Based Iterative Algorithms for Solving a Class of Matrix Equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216- 1221, Aug. 2005.

[113] R. Fa, R. C. de Lamare, "Multi-Branch Successive Interference Cancellation for MIMO Spatial Multiplexing Systems", *IET Communications*, 2011.

[114] M. Chiani, "Introducing erasures in decision-feedback equalization to reduce error propagation," *IEEE Trans. Commun.*, vol. 45, no. 7, pp. 757-760, July 1997.

[115] S.-J. Hwang and P. Schniter, "Efficient Sequence Detection of Multicarrier Transmissions over Doubly Dispersive Channels", *EURASIP Journal on Applied Signal Processing.*, Vol. 2006, Article ID. 93638, Pages 1-17, 2006.

[116] J. Salo, G.D. Galdo, P. Kyosti, M. Milojevic, D. Laselva, C. Schneider, "MATLAB implementation of 3GPP spatial channel model (3GPP TR 25.996)," January 2005, On-line: http://www.tkk.fi/units/radio/scm/.

[117] Y. H. Gan, C. Ling, W. H. Mow, "Complex Lattice Reduction Algorithm for Low-Complexity Full-Diversity MIMO Detection," *IEEE Trans. on Signal Processing,* , vol. 57, no. 7, pp. 2701-2710, July 2009.

[118] T. Minka, "The Lightspeed Matlab toolbox, efficient operations for Matlab programming, version 2.2", 17-Dec-2007, Microsoft Corperation.

[119] M. L. Honig and J. S. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Trans. Commun.*, vol. 50, pp. 986-994, June 2002.

[120] R. C. de Lamare, M. Haardt and R. Sampaio-Neto, "blind Adaptive Constrained Reduced-Rank Parameter Estimation based on Constant Modulus Design for CDMA Interference Suppression," *IEEE Transactions on Signal Processing*, vol. 56., no. 6, June 2008.

[121] R. C. de Lamare and R. Sampaio-Neto, "Reduced-rank Interference Suppression for DS-CDMA based on Interpolated FIR Filters", *IEEE Communications Letters*, vol. 9, no. 3, March 2005.

[122] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank MMSE Filtering with Interpolated FIR Filters and Adaptive Interpolators", *IEEE Signal Processing Letters*, vol. 12, no. 3, March, 2005.

[123] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Interference Suppression for DS-CDMA Systems based on Interpolated FIR Filters with Adaptive Interpolators in Multipath Channels", *IEEE Trans. Vehicular Technology*, Vol. 56, no. 6, September 2007.

[124] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank MMSE Parameter Estimation based on an Adaptive Diversity Combined Decimation and Interpolation Scheme," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, April 15-20, 2007, vol. 3, pp. III-1317-III-1320.

[125] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Adaptive Filtering Based on Joint Iterative Optimization of Adaptive Filters", *IEEE Signal Processing Letters*, Vol. 14, no. 12, December 2007.

[126] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter," *IEEE Trans. on Sig. Proc.*, vol. 49, No. 2, February, 2001.

[127] H. Qian and S.N. Batalama, "Data record-based criteria for the selection of an auxiliary vector estimator of the MMSE/MVDR filter", *IEEE Trans. on Communications*, vol. 51, no. 10, Oct. 2003, pp. 1700 - 1708.

[128] Y. Hua, M. Nikpour, and P. Stoica, "Optimal reduced-rank estimation and filtering," *IEEE Transactions on Signal Processing*, 2001, V.49, 457-469.

[129] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Adaptive Filtering Based on Joint Iterative Optimization of Adaptive Filters," *IEEE Signal Processing Letters*, Vol. 14 No. 12, December 2007, pp. 980 - 983.

[130] R. C. de Lamare and P. S. R. Diniz, "Set-Membership Adaptive Algorithms based on Time-Varying Error Bounds for CDMA Interference Suppression", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, February 2009 , pp. 644 - 654.

[131] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation and Filtering", *IEEE Transactions on Signal Processing*, vol. 57, no. 7, July 2009, pp. 2503 - 2514.

[132] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Space-Time Adaptive Interference Suppression With Joint Iterative Least Squares Algorithms for Spread-Spectrum Systems," *IEEE Transactions on Vehicular Technology*, vol.59, no.3, March 2010, pp.1217-1228.

[133] R.C. de Lamare, R. Sampaio-Neto, M. Haardt, "Blind Adaptive Constrained Constant-Modulus Reduced-Rank Interference Suppression Algorithms Based on

Interpolation and Switched Decimation," *IEEE Trans. on Signal Processing*, vol.59, no.2, pp.681-695, Feb. 2011.

[134] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Equalization Algorithms Based on Alternating Optimization Design Techniques for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol.60, no.6, pp.2482-2494, July 2011.

[135] T. Wang, R. C. de Lamare, and P. D. Mitchell, "Low-Complexity Set-Membership Channel Estimation for Cooperative Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol.60, no.6, pp.2594-2607, July 2011.