

CODING AND PROBABILISTIC INFERENCE METHODS
FOR DATA-DEPENDENT TWO-DIMENSIONAL
CHANNELS

by

Mohsen Bahrami

Copyright © Mohsen Bahrami 2019

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2019

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Mohsen Bahrami titled Coding and Probabilistic Inference Methods for Data-Dependent Two-Dimensional Channels and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Bane Vasic

Bane Vasic

Date: 6/4/2019

Ravi Tandon

Ravi Tandon

Date: 6/4/2019

Michael Marcellin

Michael Marcellin

Date: 6/4/2019

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Bane Vasic

Bane Vasic,
Dissertation Committee Chair,
Professor,
Electrical and Computer Engineering Department,
University of Arizona

Date: 6/4/2019

ARIZONA

DEDICATION

I dedicated this dissertation to my wife "Elnaz".

TABLE OF CONTENTS

| | |
|--|----|
| LIST OF FIGURES | 6 |
| LIST OF TABLES | 9 |
| ABSTRACT | 10 |
| CHAPTER 1 Introduction | 14 |
| CHAPTER 2 GBP-based TDMR Detector and Decoder | 20 |
| 2.1 TDMR System Model | 21 |
| 2.1.1 Magnetic Medium | 22 |
| 2.1.2 Write Procedure | 23 |
| 2.1.3 Read Procedure | 24 |
| 2.2 GBP-based 2-D ISI Detection | 25 |
| 2.2.1 Gibbs Free Energy and Kikuchi Approximation | 26 |
| 2.2.2 Hard Decisions from GBP | 28 |
| 2.2.3 Choice of Regions | 32 |
| 2.2.4 Soft Information from GBP | 33 |
| 2.3 Using Soft Information from GBP for Iterative Decoding of Coded TDMR Channels | 34 |
| 2.4 Lower and Upper Bounds on the SIR of Voronoi Channel | 35 |
| 2.5 Frame Error Rate Results | 37 |
| CHAPTER 3 Investigation into Harmful Patterns over Two-Dimensional Magnetic Recording | 38 |
| 3.1 Noise Characteristics of The TDMR Systems | 40 |
| 3.2 Evaluation of Utilizing Constrained Coded Data in TDMR sys- tems | 43 |
| 3.2.1 Constrained Codes for Magnetic Recording Channels | 44 |
| 3.2.2 2-D Constrained Sequence Generator | 45 |
| 3.3 Detection Scheme | 50 |
| 3.4 Evaluation of Performance of 2-D Constrained Codes in TDMR | 52 |
| CHAPTER 4 Constraint Gain for TDMR Channels | 56 |
| 4.1 Constraint Gain | 58 |
| 4.2 Max-Entropic Information Rate | 62 |
| 4.3 On the Accuracy of The GBP-based TDMR Detector and In- formation Rate Estimator | 64 |
| 4.4 TDMR 2-D Constraint Gain Results | 66 |

TABLE OF CONTENTS – *Continued*

| | | |
|------------|--|-----|
| CHAPTER 5 | Deliberate Bit Flipping Coding Scheme | 68 |
| 5.1 | Tilings and Polyominoes | 71 |
| 5.2 | Channel Model | 73 |
| 5.3 | Problem Formulation | 78 |
| 5.4 | A Probabilistic Graphical Formultion for Minimzing Bit Flips | 86 |
| 5.5 | Numerical Results | 90 |
| 5.5.1 | Statistics of The Number of Bit Flips for Removing 2-D Isolated-Bits Patterns | 91 |
| 5.5.2 | Performance Evaluation of The GBP-Guided DBF Method | 94 |
| 5.5.3 | Comparison Results on BSC | 96 |
| CHAPTER 6 | A Log-Likelihood Ratio based GBP for 2-D Channels . | 100 |
| 6.1 | Constraint Satisfiability Problem | 102 |
| 6.2 | Log-Likelihood Ratio based GBP Algorithm | 105 |
| 6.3 | Image Denoising over 2-D Gaussian Channels | 108 |
| 6.4 | Simulation Results | 110 |
| 6.5 | Comparison Results with JTED | 111 |
| CHAPTER 7 | Conclusions | 113 |
| References | | 117 |

LIST OF FIGURES

| | | |
|-----|---|----|
| 2.1 | The block diagram of TDMR system includes constrained coding, read channel and multi-track decoder. Prior to being written to the channel, user data is first encoded by a constrained code in which occurrence of harmful patterns is forbidden or suppressed (constrained coding). | 21 |
| 2.2 | An example of the Voronoi channel model. The grains on the medium are modeled as the Voronoi regions formed from the random grain centers generated using Poisson disk process. The centers are separated by at least $CTC = 10$ nm. The rectangular cells indicate the channel bits. All grains whose centers are within a bit region are polarized according to the bit value. The bit size is $TW \times BP = 30 \text{ nm} \times 15 \text{ nm}$. These parameters correspond to arbitrary but realistic physical values | 22 |
| 2.3 | Factors $f_{i,j}(\cdot)$ of a 3×3 page are shown. The corresponding region graph with all regions and sub-regions is also shown. The arrows in the region graph show the flow of messages in the GBP algorithm. | 27 |
| 2.4 | BER performance of the GBP algorithm for different choices of regions. The best performance is seen when all sub-regions of sizes 2×3 , 3×2 , 1×3 , 3×1 , 2×1 , 1×2 , 2×2 , 1×1 are chosen. Omitting any of the regions would not ensure that the beliefs marginalize to the same values in the intersection of regions. Severe degradation in performance is seen if there are large number of descendants for the omitted regions. | 33 |
| 2.5 | Lower and upper bounds on the SIR of Voronoi channel of TDMR system with the parameters given in Table 2.1 | 35 |
| 2.6 | The FER result of quasi-cyclic column weight four LDPC code with $N = 756$, $R = 0.66$ with respect to the parameter TW for the Voronoi channel. Also, the FER of BCH code (1023, 675) is plotted for reference. | 37 |
| 3.1 | Observation of media noise variance for the Voronoi channel with the parameters $CTC = 7\text{nm}$, $BP = 7.5\text{nm}$ and $TW = 16\text{nm}$. In the 3×3 input patterns 0 and 1 are represented by white and black, respectively. It is shown that the harmful patterns for the Voronoi channel with 2-D ISI are ones eliminated by the no isolated bit constraint. | 41 |
| 3.2 | Factor graph of a 4×4 variable nodes with local constraints. | 46 |

LIST OF FIGURES – *Continued*

| | | |
|-----|--|----|
| 3.3 | A region graph of a 4×4 variable nodes generated utilizing the parent to child scheme [1] | 48 |
| 3.4 | BER comparison of un-coded (TDMR(1)) and coded (TDMR(2)) systems with different bit areas and the same storage density in absence of electronic noise. Constrained coding improves the performance by avoiding the data patterns that result in high media noise | 54 |
| 3.5 | BER comparison of un-coded (TDMR(3)) and coded (TDMR(4)) systems with different bit areas and the same storage density in the presence of electronic noise. The impact of constrained coding is higher at high SNRs as the media noise dominates the electronic noise in this region. $\text{SNR}_{\text{Elec}}=10$ dB is a trade-off point where the performance gain due to constrained coding compensates the effects of both media and electronic noise. | 54 |
| 4.1 | Hard-decision detection performance of a GBP detector versus optimal (MAP) detector error probability in terms of average BER per bits as a function of TW for a TDMR system. It should be noted that the GBP curve has no markers, but the MAP performance points, represented by markers alone, fall exactly on top of the GBP lines. The standard deviation of the results is small. | 65 |
| 4.2 | The KL-distance $D(b(\mathbf{x}) p(\mathbf{x}))$ between the beliefs $b(\mathbf{x})$ computed using GBP and marginals of optimal MAP $p(\mathbf{x})$ versus TW for a TDMR-based Voronoi channel. | 66 |
| 4.3 | Estimating the constraint gain for the 2-D n.i.b. constraint over the Voronoi based TDMR channel with the parameters given in Table 4.1. | 67 |
| 5.1 | Two examples of polyominoes: (a) a 2×2 square and (b) a cross. | 72 |
| 5.2 | Figure demonstrates $\mathcal{P}_{i,j}$ over a rectangle when the polyomino is: (a) a 2×2 square and (b) a cross. | 73 |
| 5.3 | A schematic representation for the channel model is given. Passing through the channel, the color of tile $x_{i,j}$ inverts with probability α_b if the configuration of $\mathcal{P}_{i,j}$, $\mathbf{x}_{\mathcal{P}_{i,j}}$, belongs to the set of harmful patterns $\mathcal{X}_{\mathcal{P}_{i,j}}^B$, otherwise it inverts with a probability of α_g | 75 |
| 5.4 | A 7×7 binary pattern \mathbf{x} is transmitted through the channel with the set of 2-D isolated-bits patterns as the set of harmful patterns. The tiles (2, 6), (3, 5), (3, 6), (3, 7), (4, 6), (6, 7), (7, 6) and (7, 7) belong to the 2-D isolated-bits patterns. Passing through the channel, the probability of error for these tiles is α_b , and for the rest of tiles is α_g | 77 |

LIST OF FIGURES – *Continued*

| | | |
|------|---|-----|
| 5.5 | The input patterns for Example 2. We assume white tiles (zero entries) outside of each input pattern. | 84 |
| 5.6 | An approximation of the occurrence probability of bit flipping for removing the forbidden patterns by the 2-D n.i.b. constraint from random 32×32 arrays are given over 8000 trials. For this experiment, $\lambda = 0.1$ in (5.28). | 92 |
| 5.7 | BCH codes of length 1024 with different code rates are used to correct the deliberate errors introduced in random 32×32 patterns for removing 2-D isolated-bits patterns. Using the flipping probabilities in Fig. 5.6 and (5.32), the UBER is calculated for BCH codes of length 1024 with different rates (and consequently d_{\min}). | 93 |
| 5.8 | The average number of flipped bits for removing 2-D isolated-bits patterns from a random 32×32 array for different $\lambda \in \{0.04, 0.1, 0.18, 0.22, 0.26\}$ over 1000 trials versus the number of GBP iterations. | 94 |
| 5.9 | The average probability of error with and without incorporating for the cases (a) $\alpha_g = 0$ and $\alpha_b \in [0.1 : 0.1 : 1]$, and (b) $\alpha_g \in [0.001 : 0.001 : 0.01]$ and $\alpha_b = 100 \times \alpha_g$ is presented. In both cases the BCH-[1024, 728, 62] code is being used. The BER comparison results are obtained using the equations (33) and (34), and executing the GBP-guided DBF algorithm over at least 50,000 random instances of user messages. | 95 |
| 5.10 | Figure shows the BER comparison results of the DBF, bit-stuffing and row-by-row coding methods on the BSC with the cross-over probability (α). The effect of error propagation can be observed in the BER curve of bit-stuffing which shows that this method is vulnerable to channel errors. The coding rate of DBF with BCH-[1024, 923, 22] code is close to the bit-stuffing method, and the rate of DBF with BCH-[1024, 768, 54] is close to the rate of row-by-row coding method. | 98 |
| 6.1 | The factor graph for the joint probability distribution in the Eq. (6.5) is given. The set of variable nodes $\mathbf{X} = \{X_1, X_2, \dots, X_7\}$ represents the error patterns and the set of factor nodes $\mathbf{C} = \{C_1, C_2, C_3\}$ verify the syndrome constraints. | 104 |
| 6.2 | Detection performance curves of GBP for 64-bit double precision format, 24-bit fixed point LLR. | 110 |
| 6.3 | Comparison results between the proposed LLR-GBP (24-bit: 8 bits fractional and 16 bits offset intervals) and JTED. | 111 |

LIST OF TABLES

| | | |
|-----|---|----|
| 2.1 | RS _{CT} (RS _{DT}) denotes the reader response span in cross-track (down-track) dimension. All the parameters in the table are specified in nanometers. * indicates that the parameter is varied in the simulations. CTC= 7nm. | 35 |
| 3.1 | RS _{CT} (RS _{DT}) denotes the reader response span in cross-track (down-track) dimension. CTC is assumed to be 7 nanometers. All the parameters in the table are specified in nanometers. * indicates that the parameter is varied in the simulations. . . . | 53 |
| 4.1 | RS _x (RS _y) denotes the reader response span in <i>x</i> -axis and <i>y</i> -axis directions, respectively. All the parameters in the table are in nanometers. * indicates that the parameter varies in simulations. | 67 |

ABSTRACT

Recent advances in magnetic recording systems, optical recording devices and flash memory drives necessitate to study two-dimensional (2-D) coding techniques for reliable storage/retrieval of information. Most channels in such systems introduce errors in messages in response to certain data patterns, and messages containing these patterns are more prone to errors than others. For example, in a single-level cell flash memory channel, inter-cell interference (ICI) is at its maximum when 101 patterns are programmed over adjacent cells in either horizontal or vertical directions. As another example, in two-dimensional magnetic recording channels, 2-D isolated-bits patterns are shown empirically to be the dominant error event, and during the read-back process inter-symbol interference (ISI) and inter-track interference (ITI) arise when these patterns are recorded over the magnetic medium. Shannon in his seminal work, “A Mathematical Theory of Communications,” presented two techniques for reliable transmission of messages over noisy channels, namely error correction coding and constrained coding. In the first method, messages are protected via an error correction code (ECC) from random errors which are independent of input data. The theory of ECCs is well studied, and efficient code construction methods are developed for simple binary channels, additive white Gaussian noise (AWGN) channels and partial response channels. On the other hand, constrained coding reduces the likelihood of corruption by removing problematic patterns before transmission over data-dependent channels. Prominent examples of constraints include a family of binary one-dimensional (1-D) and 2-D (d, k) -run-length-limited (RLL) con-

straints which improves resilience to ISI timing recovery and synchronization for bandwidth limited partial response channels, where d and k represent the minimum and maximum number of admissible zeros between two successive ones in any direction of array. In principle, the ultimate coding approach for such data-dependent channels is to design a set of sufficiently distinct error correction codewords that also satisfy channel constraints. Designing channel codewords satisfying both ECC and channel constraints is important as it would achieve the channel capacity. However, in practice this is difficult, and we rely on sub-optimal methods such as forward concatenation method (standard concatenation), reverse concatenation method (modified concatenation), and combinations of these approaches. In this dissertation, we focus on the problem of reliable transmission of binary messages over data-dependent 2-D communication channels. Our work is concerned with several challenges in regard to the transmission of binary messages over data-dependent 2-D channels.

1. Design of Two-Dimensional Magnetic Recording (TDMR) Detector and Decoder: TDMR achieves high areal densities by reducing the size of a bit comparable to the size of the magnetic grains resulting in 2-D ISI and very high media noise. Therefore, it is critical to handle the media noise along with the 2-D ISI detection. In this work, we tune the Generalized Belief Propagation (GBP) algorithm to handle the media noise seen in TDMR. We also provide an intuition into the nature of hard decisions provided by the GBP algorithm.
2. Investigation into Harmful Patterns for TDMR channels: This work investigates into the Voronoi based media model to study the harmful patterns over multi-track shingled recording systems. Through realistic quasi micromagnetic simulations studies, we identify 2-D data patterns

that contribute to high media noise. We look into the generic Voronoi model and present our analysis on multi-track detection with constrained coded data. We show that 2-D constraints imposed on input patterns result in an order of magnitude improvement in the bit error rate for TDMR systems.

3. **Understanding of Constraint Gain for TDMR Channels:** We study performance gains of constrained codes in TDMR channels using the notion of constraint gain. We consider Voronoi based TDMR channels with realistic grain, bit, track and magnetic-head dimensions. Specifically, we investigate the constraint gain for 2-D no-isolated-bits constraint over Voronoi based TDMR channels. We focus on schemes that employ the GBP algorithm for obtaining information rate estimates for TDMR channels.
4. **Design of Novel Constrained Coding Methods:** In this work, we present a deliberate bit flipping (DBF) coding scheme for binary 2-D channels, where specific patterns in channel inputs are the significant cause of errors. The idea is to eliminate a constrained encoder and, instead, embed a constraint into an error correction codeword that is arranged into a 2-D array by deliberately flipping the bits that violate the constraint. The DBF method relies on the error correction capability of the code being used so that it should be able to correct both deliberate errors and channel errors. Therefore, it is crucial to flip minimum number of bits in order not to overburden the error correction decoder. We devise a constrained combinatorial formulation for minimizing the number of flipped bits for a given set of harmful patterns. The GBP algorithm is used to find an approximate solution for the problem.

5. Devising Reduced Complexity Probabilistic Inference Methods: We propose a reduced complexity GBP that propagates messages in Log-Likelihood Ratio (LLR) domain. The key novelties of the proposed LLR-GBP are: *(i)* reduced fixed point precision for messages instead of computational complex floating point format, *(ii)* operations performed in logarithm domain, thus eliminating the need for multiplications and divisions, *(iii)* usage of message ratios that leads to simple hard decision mechanisms.

CHAPTER 1

Introduction

Machine learning techniques have gained attention recently in communications [2], signal processing [3], and error-correction coding [4] for predictive inference tasks. Many of these inference problems can be reformulated as the computation of marginal probabilities of a joint probability distribution over the set of solutions of a constraint satisfaction problem (CSP) [5, 6]. A CSP consists of a number of variables and a number of constraints, where each constraint specifies admissible values of a subset of variables. A solution to a CSP is an assignment of variables satisfying all the constraints. Message passing algorithms have been successfully used for solving hard CSPs [7]. Traditional low-complexity approximate algorithms for solving these problems are based on belief propagation (BP) [8, 9] which operate on factor graphs. BP, as an algorithm to compute marginals over a factor graph, has its roots in the broad class of Bayesian inference problems [10]. It is well known that the BP algorithm gives exact inference only on cycle-free graphs (trees). It has been also observed that in some applications BP surprisingly can provide close approximations to exact marginals on loopy graphs. However, an understanding of the behavior of BP in the latter case is far from complete. Moreover, it is known that BP does not perform well on graphs which contain a large number of short cycles. The validity of BP algorithm for computing marginal probability distributions relies on the assumption that messages sent over factor graph into a node from its neighboring nodes are independent. In factor graphs with cycles, failures of BP algorithm show the existence of correlation among

messages. Statistical physicists attribute these correlations among messages over a loopy factor graph (a factor graph with cycles) to the geometry of the solution space of CSPs. The density of constraint is determined by $\alpha = \frac{M}{N}$ and this parameter identifies satisfiability thresholds for the solution space of CSPs [11–15]. As $N \rightarrow \infty$, a CSP becomes less likely to be satisfiable as α grows. We assume that there exists a SAT threshold α_C for a given CSP. At fixed α when $N \rightarrow \infty$, a CSP is almost surely satisfiable if $\alpha < \alpha_C$, and the problem is almost surely un-satisfiable if $\alpha > \alpha_C$. In statistical physics, there is an assumption on existence of a critical value α_d for constraint density, which is smaller than the threshold density α_C , at which the structure of the solution space changes. Below the critical value, a CSP has exponentially many solutions which form a big cluster and the Hamming distance of solutions are very small [16]. However, close to the critical threshold, the solution space consists of many smaller clusters and the solutions are far apart. Each cluster has its local minimas such that there exist exponentially many widely separated solutions. These local minimas can be traps for local search algorithms, like BP algorithm.

The Survey Propagation (SP) algorithm is proposed to find satisfying solutions for highly dense constraint density and large instances of random K -SAT problems around the critical value. A random K -SAT refers to a satisfiability problem with a set of variables and a set of clauses (with Boolean functions) in which each clause contains K literals. K -SAT problems have been shown to be NP-complete for $K \geq 3$ [17]. The SP has its origin in statistical physics based on the cavity method [18] and has been shown to deal with the clustering phenomenon of solution space for large instances K-SAT problems at much higher densities than previous methods [19]. In the original derivation of SP algorithm, the messages are sent among clusters in the solution space of CSPs,

which provides information about the fraction of solutions (assignments) in a cluster in which given variables are frozen or free. The SP's updates can be obtained from BP with an extended variable space $\{0, 1, \star\}$, where \star or joker state represents the state of variables which are free in a cluster of solution space. Experimental studies show that SP is more efficient than BP for random SAT problems [20]. A new class of message-passing algorithm called generalized belief propagation (GBP) is introduced in [1] to solve the problem of computing marginal probability distributions on factor graphs with short cycles. The algorithm relies on the extension of cluster variation method [21, 22], which is called the region graph method. The GBP algorithm provides approximate marginals by minimizing the Gibbs free energy using region graph method. In GBP, messages are sent among clusters of variables nodes instead of the node-to-node message passing fashion in BP and SP. GBP algorithm is used over dense graphs for detection and information rate estimation for two-dimensional (2-D) inter-symbol interference and Gaussian channels [23, 24]. Furthermore, GBP has been successfully employed for decoding of classical and quantum LDPC codes on sparse graphs with short cycles [25, 26]. More recently GBP has been shown empirically to have good performance, in either accuracy or convergence properties, for certain applications [24, 27].

In this dissertation, we focus on the problem of reliable transmitting binary messages over data-dependent communication channels and recovering them back at the receiver side. This problem is one of the most fundamental problems in communication theory, and can be considered as an instance of a CSP. Shannon in his seminal work [28] introduced two coding schemes for reliable transmission of information over noisy channels, namely error correction coding and constrained coding. The first method protects user messages against random errors, which are independent of input data, by introducing

redundancy in the messages prior to transmission. On the other hand, a constrained coding method assumes that channel solely introduces errors in response to specific patterns in input messages, and removing these problematic patterns makes the channel noiseless. We consider the following challenges in regard to reliable transmission of binary messages over data-dependent 2-D channels, which include, but not limited to, *(i)* design of novel error correction and constrained coding techniques, *(ii)* use of state-of-the-art message-passing algorithms for probabilistic inference, and *(iii)* devising reduced complexity 2-D detection and decoding methods. The organization of the dissertation is as follows:

In Chapter 2, we propose a method to handle the media noise seen in a TDMR channel, as an example of a data-dependent 2-D channel, using the Generalized Belief Propagation (GBP) based detector. We use the GBP algorithm for signal detection in conjunction with a Belief Propagation (BP) algorithm for Low-Density Parity-Check (LDPC) decoding. We give an insight into the nature of signal classification (hard decisions) by GBP to be motivated towards minimizing frame-error-rate. We also evaluate the performance of the GBP algorithm for different choices of regions suitable for TDMR. The GBP algorithm can be formulated to handle correlation in the media noise and exchange information in a turbo fashion with the BP algorithm for further gains in the TDMR performance.

We study the pattern dependent characteristics of media noise in TDMR using a Voronoi media model in Chapter 3. We identify the no-isolated-bits constraint that reduces the impact of media noise. We study the performance of the constrained coding using a BCJR based multi-track detector. When the media noise is high compared to the electronic noise, the rate loss due to constrained coding is compensated by the performance gains when compared

against uncoded systems with the same storage density. We also introduce the main idea of our method for generating 2D constrained sequences based on the GBP algorithm.

In Chapter 4, we investigate performance gains of incorporating constrained codes in Two dimensional Magnetic Recording (TDMR) channels using the notion of *constraint gain*. A Voronoi based TDMR channels with realistic grain, bit, track and magnetic-head dimensions is considered as the TDMR channel model. We focus on 2-D n.i.b. constraint for the Voronoi based TDMR channels. We focus on schemes that employ the generalized belief propagation algorithm for obtaining information rate estimates for TDMR channels.

In Chapter 5, we propose a coding scheme for data-dependent 2-D channels which is based on a deliberate bit flipping method. Deliberate errors are introduced into an error correction codeword which is arranged into a 2-D array to remove harmful patterns before transmission. The technique relies on the error correction capability of the code being used, and the number of deliberate errors should be small enough not to overburden the error correction decoder. In this chapter, we focus on minimizing the number of deliberate errors in the DBF scheme for removing a set of given configurations from input patterns. We devise a probabilistic graphical model for the minimization problem by reformulating it as a 2-D MAP problem. We use the GBP algorithm to find an approximate solution for the 2-D MAP formulation of the problem. Statistics of the number of bit flips for removing 2-D isolated-bits patterns are extracted, and we show that how these numbers are comparable with the error correction capability of BCH codes being used. Furthermore, we investigate the suitability of DBF method for imposing 2-D constraint over a BSC against classical constrained coding methods which suffer from error propagation.

In Chapter 6, we propose a LLR version in order to reduce both the computational complexity and the storage requirements for GBP. From a computational perspective, the main advantages of the proposed approach are:

1. arithmetic operations are performed in fixed point formats rather than computationally complex floating point formats,
2. multiplications in the belief and message update rules are reduced to additions,
3. divisions in the message update rules are reduced to subtractions, and
4. signed based hard-decision extraction mechanism for single variable regions, as is the case in the vast majority of detection problems.

Regarding the approximation of the logarithm of the addition, our approach employs a maximum computation, as well as comparisons with a number of offline computed constants. Therefore, the proposed LLR version of GBP employs only fixed point addition based operations - addition, subtraction and comparisons – that makes it suitable for hardware acceleration on FPGA devices.

CHAPTER 2

GBP-based TDMR Detector and Decoder

Two dimensional magnetic recording (TDMR) is a promising technology to increase the areal densities beyond 800 Gb/in² using sophisticated signal processing algorithms on the currently available magnetic medium by reducing the track width. The signal processing algorithms in TDMR have to handle the 2-D ISI and very high media noise arising due to irregularities in the medium.

The correlation and data dependent nature of the media noise can be used to reduce the effect of media noise on the signal processing algorithms in TDMR. Khatami and Vasić [29] have used constrained codes along with GBP detector to avoid harmful patterns that contribute to high media noise. Matcha and Srinivasa [30] have used pattern dependent noise prediction filters along with a 2-D soft-output Viterbi algorithm (2-D SOVA) to handle the media noise.

We use the GBP algorithm for signal detection. The GBP algorithm a graph based iterative algorithm where the messages are passed across regions instead of between nodes as seen in the BP algorithm [1]. The performance of the algorithm in relation to the MAP/ML criteria and the optimal choice of regions is not well understood. In this chapter, we model the media noise from a Voronoi based media model as a pattern dependent noise. We formulate the GBP algorithm to handle the media noise and obtain soft-outputs useful to decode a LDPC code. We also provide intuition into the nature of hard decisions given by GBP by looking at the GBP as a convex optimization problem.

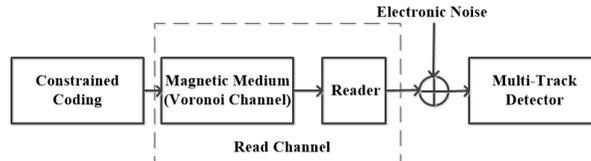


Figure 2.1: The block diagram of TDMR system includes constrained coding, read channel and multi-track decoder. Prior to being written to the channel, user data is first encoded by a constrained code in which occurrence of harmful patterns is forbidden or suppressed (constrained coding).

The chapter is organized as follows: In Section 2.1, we describe the Voronoi based TDMR channel model. In Section 2.2, we provide insights into the nature of hard decisions from GBP algorithm and formulate the algorithm to handle media noise. In Section 2.3, we use GBP algorithm to bound the TDMR channel capacity for designing the LDPC code of appropriate rate. We also discuss the numerical results where LDPC codes are decoded using soft outputs from the GBP algorithm in Section 2.5.

2.1 TDMR System Model

The study of the effects of jitter noise on the signal processing algorithms in TDMR systems requires sophisticated channel models that include the random grain distribution on the recording medium. Fig. 2.1 provides a block diagram of the TDMR system utilized in this chapter. We model the TDMR channel using a Voronoi model [31] where each grain is specified by a Voronoi region. 2-D constrained sequences from the input alphabet $\mathcal{X} = \{-1, +1\}$ are written on the magnetic medium. Without loss of generality -1 and $+1$ denote the bits 0 and 1 respectively. A magnetic reader is utilized to read data written on the Voronoi channel, and produces symbols from the alphabet $\mathcal{Y} = \mathbb{R}$. The electronic noise is modeled by an Additive White Gaussian Noise (AWGN) with variance σ_e^2 . The noisy output is equalized and detected using a multi-track detector in order to retrieve the symbols written on the Voronoi channel.

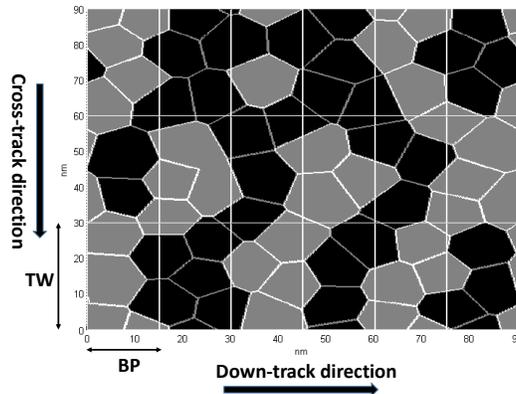


Figure 2.2: An example of the Voronoi channel model. The grains on the medium are modeled as the Voronoi regions formed from the random grain centers generated using Poisson disk process. The centers are separated by at least $CTC = 10$ nm. The rectangular cells indicate the channel bits. All grains whose centers are within a bit region are polarized according to the bit value. The bit size is $TW \times BP = 30$ nm \times 15 nm. These parameters correspond to arbitrary but realistic physical values

In this section, we introduce the details of the model used in this chapter.

TDMR channel models typically involve three components: a) media model: models the distribution of grains on the medium b) write-head procedure: models the magnetization process of grains while writing data on to the Voronoi channel and c) read-head procedure: models the readback signal. For the sake of completeness we give these models as described in [29].

2.1.1 Magnetic Medium

In TDMR systems, a grain is the smallest region that is uniformly magnetized. A Voronoi model is utilized to simulate the non-ideal features of the magnetic medium [31]. A Voronoi region \mathcal{S} with a center c is the collection of points on a 2-D (Euclidean) plane that are closer to the center c than to any other grain center. The points on the boundary of a Voronoi region are equidistant from their two closest centers. In this model, the medium is visualized as a random tiling of Voronoi regions where each Voronoi region represents a grain on the medium. There is more than one way to generate a random Voronoi

tiling of a plane. In this chapter, the grain centers are generated according to the Poisson-disk distribution with boundary sampling introduced in [32]. The Poisson-disk distribution is characterized by the center-to-center (CTC) distance, the minimum permissible distance between any two grain centers. In this method, the grain centers are not allowed to be closer than the (CTC) distance and there is at least one grain center at this distance. The (CTC) distance determines the size and shape of grains. In the following, the Voronoi channel parameters are introduced.

A rectangular grid is defined on the medium, where each rectangular cell corresponds to a channel bit and is characterized by

- Bit Period (BP): the length of each bit in the down-track direction.
- Track-Width (TW): the length of each bit in the cross-track direction.

An example of the TDMR channel generated based on the Voronoi model is given in Fig. 2.2.

2.1.2 Write Procedure

Constrained sequences are written on the Voronoi channel at this step. The channel input signal $x(t_1, t_2)$ is defined by

$$x(t_1, t_2) = \sum_i \sum_j x_{i,j} \Pi_{TW}(t_1 - i \times TW) \Pi_{BP}(t_2 - j \times BP),$$

where $x_{i,j} \in \mathcal{X}$ is the symbol which will be written on the $(i, j)^{\text{th}}$ bit area and

$$\Pi_T(t) = \begin{cases} 1, & 0 \leq t < T, \\ 0, & \text{otherwise.} \end{cases}$$

In TDMR systems, the write head procedure does not have any *a-priori* knowledge of the grain shapes, sizes and positions on the magnetic medium. Therefore, the bit areas are considered to be in the form of rectangles. The write head induces a magnetization pattern on the track directly below its head at the center of each rectangular cell such that all grains whose centers are within the bit area are polarized according to the value of $x_{i,j}$.

2.1.3 Read Procedure

We model the read-head response to be a 2-D Gaussian pulse with a span of three bit areas in both directions. The 2-D Gaussian pulse is characterized by the pulse widths PW_{50} and TW_{50} at half-amplitude in the down-track and cross-track directions, respectively. We suppose that the read-head picks up magnetization only from $m \times n$ cells. As a result, the read-head output sample $y_{i,j}$ at the center of the $(i, j)^{\text{th}}$ cell depending only on the polarity of the grains in the $m \times n$ neighborhood around the $(i, j)^{\text{th}}$ cell, denoted as $C_{i,j}$. The read-head parameters are chosen such that the ISI span does not exceed 3×3 bit areas throughout the simulations, i.e., $m = n = 3$.

Let $s_{i,j} \in \mathbb{R}$ be the read-back signal samples of the ideal magnetic medium, where the bit areas considered to be rectangular, and $y_{i,j} \in \mathbb{R}$ be the read-back signal samples of the non-ideal medium for the bit cell (i, j) . The read-back signal of ideal medium, $s_{i,j}$, is obtained by convolving the magnetization pattern of ideal medium with the read-head impulse response $h(t_1, t_2)$ and sampling at each center of bit area in the down-track direction. We consider that the read-head impulse response of 3×3 span. Therefore, the read-back signal of ideal medium can be written as

$$s_{i,j} = \sum_{k_1=-1}^{+1} \sum_{k_2=-1}^{+1} x_{i-k_1, j-k_2} h_{k_1, k_2}, \quad (2.1)$$

where h_{k_1, k_2} is the sampled output of impulse response of read-head,

$$h_{k_1, k_2} = \iint_{\mathcal{A}_{k_1, k_2}} h(t_1, t_2) dt_1 dt_2, \quad (2.2)$$

that \mathcal{A}_{k_1, k_2} is the rectangular area of bit (k_1, k_2) . In order to model the effect of irregular boundaries on the read-back signal of ideal magnetic medium, we define the media noise $n_{i,j}$ as an additive noise which is dependent on each 3×3 span of input data, the coded signal which is written on the Voronoi channel. Any change in the read-back signal due to the shift in the grain-boundaries is considered as media noise. This depends not only on the regions of the grains in $C_{i,j}$, but also on their polarities. Therefore, this noise is correlated in both down-track and cross-track directions and is data-dependent. Thus, we incorporate the effect of media noise to the read-back signal of ideal medium, $s_{i,j}$, in the following form

$$y_{i,j} = s_{i,j} + n_{i,j}, \quad (2.3)$$

where $y_{i,j}$ is the noisy read-back signal sample for the $(i, j)^{\text{th}}$ cell.

2.2 GBP-based 2-D ISI Detection

Generalized belief propagation (GBP) algorithm is a graph based decoding/detection algorithm that can be formulated as a convex optimization problem that minimizes the Gibbs free energy [1]. The algorithm provides a method

to approximate marginal distributions which makes it suitable for MAP detection with soft outputs.

The GBP algorithm is known to give exact marginals if and only if the region based graph has no loops [33]. Even though the region based graphs always contain loops when used for 2-D ISI signal detection, the GBP algorithm provides a method to approximate the marginals that are empirically observed to be close to the actual marginals.

In this section, we provide insights into the nature of hard decisions from the GBP algorithm and evaluate the performance of the GBP algorithm over a chosen 2-D ISI channel for different choices of region. We next formulate the GBP algorithm for the noise characteristics seen in the Voronoi based TDMR channel model.

2.2.1 Gibbs Free Energy and Kikuchi Approximation

Assuming uniform distribution of the input bits and white noise samples in the channel model, the *a-posteriori* probability of \mathbf{x} given read-back samples \mathbf{y} is given by

$$\begin{aligned} p(\mathbf{x} | \mathbf{y}) &= p(\mathbf{y} | \mathbf{x}) p(\mathbf{x}) p(\mathbf{y})^{-1} \propto p(\mathbf{y} | \mathbf{x}) \\ p(\mathbf{y} | \mathbf{x}) &= \prod_{i,j} f_{i,j}(\mathbf{x}_{i,j}) \end{aligned} \quad (2.4)$$

where $f_{i,j}(\mathbf{x}_{i,j}) = p(y_{i,j} | \mathbf{x}_{i,j})$ is the distribution function of noise sample at location (i, j) . Therefore, we have

$$p(\mathbf{x} | \mathbf{y}) = \frac{1}{Z} \prod_{i,j} f_{i,j}(\mathbf{x}_{i,j}), \quad (2.5)$$

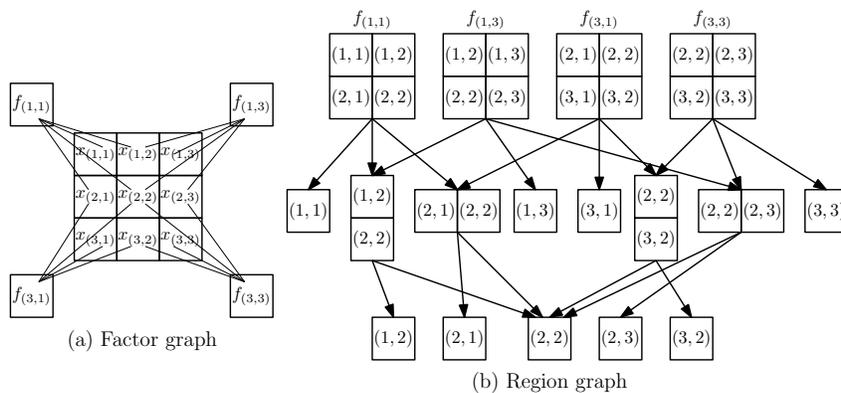


Figure 2.3: Factors $f_{i,j}(\cdot)$ of a 3×3 page are shown. The corresponding region graph with all regions and sub-regions is also shown. The arrows in the region graph show the flow of messages in the GBP algorithm.

for some $Z(\mathbf{y})$. Let $b(\mathbf{x})$ represent the belief of the *a-posterior* probability (APP). From the properties of KL-divergence, the belief $b(\mathbf{x}) = p(\mathbf{x} | \mathbf{y})$ can be achieved by minimizing the free energy given by

$$F = E - H = \mathcal{D}(b(\mathbf{x}) \| p(\mathbf{x} | \mathbf{y})) - \ln Z(\mathbf{y}), \quad (2.6)$$

$$\text{average energy } E = - \sum_{i,j} \sum_{\mathbf{x}_{i,j}} b(\mathbf{x}_{i,j}) \ln f_{i,j}(\mathbf{x}_{i,j}), \quad (2.7)$$

$$\text{entropy } H = \sum_{\mathbf{x}} b(\mathbf{x}) \ln b(\mathbf{x}). \quad (2.8)$$

Let a region $R \subset \mathbb{R}^2$ be defined as a set of positions within a page. Let \mathcal{R} represent a collection of such regions such that each of $\mathbf{x}_{i,j}$ is included in at least one region. For each $R \in \mathcal{R}$, let \mathbf{x}_R be the vector of bits in the region R and $b(\mathbf{x}_R)$ and $p(\mathbf{x}_R)$ be the corresponding marginal beliefs and probabilities. The regions are partially ordered based on the containment of one region inside another [1]. A region graph is formed using this partial ordering as shown in Figure 2.3.

The free energy is approximated using the entropy of individual regions as

$$\begin{aligned} \hat{F} &= - \sum_{i,j} \sum_{\mathbf{x}_{i,j}} b(\mathbf{x}_{i,j}) \ln f_{i,j}(\mathbf{x}_{i,j}) \\ &\quad + \sum_{R \in \mathcal{R}} c_R \sum_{\mathbf{x}_R} b(\mathbf{x}_R) \ln b(\mathbf{x}_R), \end{aligned} \quad (2.9)$$

where c_R are overcounting numbers defined as $c_R = \sum_{p \in \mathcal{P}_R} 1 - c_p$ and \mathcal{P}_R are parents of region R in the region graph. This approximation is called Kikuchi approximation or region based approximation (RBA). The marginals $b(\mathbf{x}_R)$ are estimated by minimizing (2.9) under the constraints

$$\sum_{u \in \mathbf{x}_p \setminus R} b(\mathbf{x}_p) = b(\mathbf{x}_R) \quad \forall p \in \mathcal{P}_R, \forall R \in \mathcal{R}. \quad (2.10)$$

These constraints ensure that the beliefs of sub-regions are obtained by marginalizing the beliefs of their parents [33]. The message update rules of GBP algorithm are obtained from the constrained optimization of \hat{F} using Lagrange multipliers.

The regions and c_R are chosen to 1) ensure unique solution to the for GBP algorithm, 2) closely approximate the marginals 3) reduce computational complexity.

2.2.2 Hard Decisions from GBP

The analysis on GBP in the literature is focused on closely approximating the marginals (soft decisions). However, the nature of hard decisions is also of interest while analyzing GBP as a signal detection/decoding algorithm. In this subsection, we use the ideas of linear programming and convex optimization to provide an insight into the behavior of GBP algorithm for hard decisions decoding/detection.

Hard decision decoding is a signal classification problem where the received signal is classified based on the decision regions within a signal space. We define the spaces of interest and the corresponding decision regions as follows.

Let $\{0, 1\}^N = \{\mathbf{m}_0, \mathbf{m}_1, \dots, \mathbf{m}_{2^N-1}\}$ represent the set of states taken by \mathbf{x} , where $N = mn$ is the number of bits in a page, $m_k(i, j)$ is the value of $x_{i,j}$ when $\mathbf{x} = \mathbf{m}_k$. Let $\mathbf{m}_i(R)$ be the vector of bits in \mathbf{m}_i restricted to the region $R \in \mathcal{R}$.

The RBA reduces the optimization problem in $b(\mathbf{x})$ space to the optimization in a lower dimensional space of marginals $\{b(\mathbf{x}_R)\}_{R \in \mathcal{R}}$. Let \mathbf{b} be the vector of beliefs $b(\mathbf{x})$, $\mathbf{x} = \mathbf{m}_0 \cdots \mathbf{m}_{2^N-1}$, and let $\mathbf{b}_{\mathcal{R}}$ represent the vector of marginals $b(\mathbf{x}_R)$, $\mathbf{x}_R = \mathbf{m}_0(R) \cdots \mathbf{m}_{2^N-1}(R)$, $R \in \mathcal{R}$. We define the space of probabilities and marginals as follows.

Definition 1 Probability space: *We define Δ as the space of probabilities/beliefs $\mathbf{b}(\mathbf{x})$ with the constraints*

$$0 \leq b(\mathbf{x} = \mathbf{m}_i) \leq 1 \text{ and } \sum_{i=0}^{2^N-1} b(\mathbf{x} = \mathbf{m}_i) = 1.$$

Let $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{2^N-1}\}$ represent the vertices of the space where \mathbf{v}_i represents $b(\mathbf{x} = \mathbf{m}_i) = 1$, $i = 0, \dots, 2^N - 1$. Let \mathbf{u} be the uniform distribution.

Marginal space: *Let Δ_M be the space of marginals $\mathbf{b}_{\mathcal{R}}(\mathbf{x})$ for the regions in \mathcal{R} with the constraints $0 \leq b(\mathbf{x}_R) \leq 1$ and $\sum_{\mathbf{x}_R} b(\mathbf{x}_R) = 1 \forall R \in \mathcal{R}$. We further enforce following constraints such that the marginals of two overlapping regions are consistent:*

$$\sum_{\mathbf{x}_{R_i \setminus R_j}} b(\mathbf{x}_{R_i}) = \sum_{\mathbf{x}_{R_j \setminus R_i}} b(\mathbf{x}_{R_j}) \quad \forall R_i, R_j \in \mathcal{R}. \quad (2.11)$$

Remark: *We can define a linear map $\mathcal{L} : \Delta \rightarrow \Delta_M$ using the marginal-*

ization operations $b(\mathbf{x}_R) = \sum_{\mathbf{x}_{R^c}} b(\mathbf{x}), R \in \mathcal{R}$.

Let $\hat{\mathbf{v}}_i = \mathcal{L}(\mathbf{v}_i), i = 0, \dots, 2^N - 1$ and $\hat{\mathbf{u}} = \mathcal{L}(\mathbf{u})$.

Definition 2 Pseudo-marginal space: Let Δ_P be a space of pseudo-marginals $\mathbf{b}_{\mathcal{R}}(\mathbf{x})$ for regions in \mathcal{R} with the constraints $0 \leq b(\mathbf{x}_R) \leq 1$ and $\sum_{\mathbf{x}_R} b(\mathbf{x}_R) = 1 \forall R \in \mathcal{R}$. These are pseudo marginals as we ignored the constraints in (2.11). Therefore, $\Delta_M \subset \Delta_P$.

Definition 3 Optimal hard decisions: The word \mathbf{m}_{FER} is said to be frame error rate (FER) optimal hard decision if

$$p(\mathbf{x} = \mathbf{m}_{FER}) > p(\mathbf{x} = \mathbf{m}_j), \quad \forall \mathbf{m}_j \neq \mathbf{m}_{FER}. \quad (2.12)$$

The word \mathbf{m}_{BER} is bit error rate (BER) optimal decision if

$$p(x_{i,j} = \mathbf{m}_{BER}(i,j)) > 0.5, \quad \forall x_{i,j}. \quad (2.13)$$

Let $\mathbf{v}_{FER}, \mathbf{v}_{BER}$ (and $\hat{\mathbf{v}}_{FER}, \hat{\mathbf{v}}_{BER}$) be the vertices in Δ (and Δ_M) corresponding to $b(\mathbf{m}_{FER}) = 1$ and $b(\mathbf{m}_{BER}) = 1$. Since the inner product $\langle \mathbf{v}_i, \mathbf{b} \rangle = b(\mathbf{x} = \mathbf{m}_i)$, the FER decision region can be written using (2.12) as

$$D_{(FER)} = \Delta \cap \bigcap_{j: \mathbf{m}_j \neq \mathbf{m}_{FER}} \{ \langle \mathbf{v}_{FER} - \mathbf{v}_j, \mathbf{b} \rangle \geq 0 \} \quad (2.14)$$

It is easy to see that all FER decision regions corresponding to each word \mathbf{m}_i intersect at \mathbf{u} . Proposition 1 identifies the FER decision region in Δ_P corresponding to (2.14).

Proposition 1 The optimal FER decision region in the pseudo marginal space Δ_P is

$$\hat{D}_{FER} = \Delta_P \cap \bigcap_{j: \hat{\mathbf{v}}_j \neq \hat{\mathbf{v}}_{FER}} \{ \langle \hat{\mathbf{v}}_{FER} - \hat{\mathbf{v}}_j, \mathbf{b}_{\mathcal{R}} \rangle \geq 0 \}. \quad (2.15)$$

Let $\tilde{D}_{FER} = \mathcal{L}(D_{FER})$ be the linear map of D_{FER} from Δ to Δ_M . Note that \tilde{D}_{FER} also has linear decision boundaries and hence the decision boundaries in Δ_P are also linear. Notice that D_{FER} and \hat{D}_{FER} are Voronoi regions in their own spaces.

Each vertex of D_{FER} is obtained as follows: Choose any subset set of points $\mathcal{V} \subseteq \{\mathbf{v}_i \mid i = 1, \dots, 2^{N-1}\}$. Centroid of points $\mathcal{V} \cup \{\mathbf{v}_{FER}\}$ is a vertex of D_{FER} .

Similarly, the vertices of \hat{D}_{FER} in (2.15) are the centroids of a subset of points from $\{\hat{\mathbf{v}}_i \neq \hat{\mathbf{v}}_{FER}\}$ and $\hat{\mathbf{v}}_{FER}$. Since the map from \mathbf{v}_i to $\hat{\mathbf{v}}_i$ is linear, the same linear map maps the centriods in Δ to centriods in Δ_P . Therefore, the vertices of \hat{D}_{FER} in (2.15) are a map of vertices of D_{FER} . Therefore, \hat{D}_{FER} in (2.15) is the optimal FER decision region in Δ_P .

The following proposition proves a property of the average energy in Δ_P that helps us in understanding the nature of signal classification by GBP.

Proposition 2 *In the pseudo marginal space Δ_P , the average energy term in the Kikuchi approximation of free energy has a constant gradient $\mathbf{g} = \frac{\partial E}{\partial \mathbf{b}_{\mathcal{R}}}$ satisfying*

$$\langle \hat{\mathbf{v}}_{FER} - \hat{\mathbf{v}}_i, \mathbf{b}_{\mathcal{R}} \rangle \leq 0 \forall \hat{\mathbf{v}}_i \neq \hat{\mathbf{v}}_{FER}, i = 0, \dots, 2^N - 1. \quad (2.16)$$

The gradient of E has the terms $-\log f_{i,j}(\mathbf{x}_{i,j})$ and hence is constant. Since $E = 0$ when $\mathbf{b}_{\mathcal{R}} = \mathbf{0}$, we can write $E(\mathbf{b}_{\mathcal{R}}) = \langle \mathbf{b}_{\mathcal{R}}, \mathbf{g} \rangle$. From (2.7), E is linear in Δ and the minima of E occurs on the boundaries of Δ . We can easily verify that E is minimized in Δ when $b(\mathbf{x} = \mathbf{m}_{FER}) = 1$ i.e., at the point $\mathbf{b} = \mathbf{v}_{FER}$. Consider the polytope in Δ_P formed by the points $\hat{\mathbf{v}}_i, i = 0, \dots, 2^N - 1$. Using the exactness of the average energy in RBA [1], we can claim that the average energy is minimum at $\mathbf{b}_{\mathcal{R}} = \hat{\mathbf{v}}_{FER}$ inside this polytope i.e., $E(\hat{\mathbf{v}}_{FER}) \leq E(\hat{\mathbf{v}}_i) \implies \langle \hat{\mathbf{v}}_{FER} - \hat{\mathbf{v}}_i, \mathbf{b}_{\mathcal{R}} \rangle \forall \hat{\mathbf{v}}_i \neq \hat{\mathbf{v}}_{FER}$.

Nature of signal classification by GBP The approximated entropy \hat{H} has maximum value at uniform distribution $\hat{\mathbf{u}}$. Proposition 2 shows that the gradient of $-E$ has the largest component along the direction of $\hat{\mathbf{v}}_{FER}$ than in the direction of any other $\hat{\mathbf{v}}_i$. Therefore, in the optimization problem to maximize $-E + H$, the component $-E$ shifts the maxima of \hat{H} closer to $\hat{\mathbf{v}}_{FER}$ i.e., within the region \hat{D}_{FER} . This shows that the inherent nature of signal classification achieved by GBP is towards optimizing FER. Due to this nature of signal classification, the GBP algorithm is suitable for hard decision decoding of error correcting codes (ECC) where FER has to be minimized. A good approximation of entropy will provide a closer approximation of the marginals resulting in optimal BER. Therefore, a good approximation of entropy is the key for the problems where BER has to be minimized.

2.2.3 Choice of Regions

In this subsection, we focus on choosing regions suitable for 3×3 ISI span. The optimal choice of regions is not trivial. Welling [34] has proposed a region pursuit algorithm based on his observations on splitting and merging of regions. However, the choice of regions larger than 3×3 is computationally prohibitive for signal detection in TDMR.

Therefore, we restrict our search to regions of size 3×3 or smaller. Let $\mathcal{R}_{p \times q}$ denote the set of regions of size $p \times q$ within a frame. The valid sizes of sub-regions are 2×3 , 3×2 , 1×3 , 3×1 , 2×1 , 1×2 , 2×2 , 1×1 . Let \mathcal{R}' denote the collection of all 3×3 regions and sub-regions within a frame.

The choice $\mathcal{R} = \mathcal{R}'$ is shown in [33] to ensure several desirable properties for the convex optimization problem:

1. The constraints in (2.10) ensure that the beliefs of regions are marginals of a distribution if and only if $\mathcal{R} = \mathcal{R}'$.

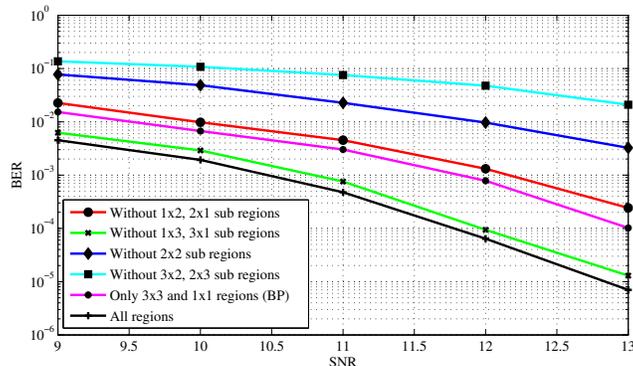


Figure 2.4: BER performance of the GBP algorithm for different choices of regions. The best performance is seen when all sub-regions of sizes 2×3 , 3×2 , 1×3 , 3×1 , 2×1 , 1×2 , 2×2 , 1×1 are chosen. Omitting any of the regions would not ensure that the beliefs marginalize to the same values in the intersection of regions. Severe degradation in performance is seen if there are large number of descendants for the omitted regions.

2. The choice achieves totally balanced condition that helps in removing bias in the approximation of entropy.
3. The choice ensures unique solution.

Figure 2.4 shows the performance of the GBP algorithm for different choices of regions over a 2-D-ISI AWGN channel given in [35]. The BER is estimated by detecting pages of size 32×32 at a time. We notice that the best performance is obtained when the $\mathcal{R} = \mathcal{R}'$. The performance is about 0.2 dB better than JTED [35] for the same 2-D ISI channel operating on 64×64 pages. We also notice severe degradation in performance if the omitted set of regions has a large number of descendants.

2.2.4 Soft Information from GBP

As discussed in Section 2.2.3, the GBP algorithm can be used to compute the *a-posteriori* probabilities of the bits. In the following, we formulate the problem of extracting soft information from the Voronoi based TDMR channel

as an instance of 2-D ISI channels.

The APP ratios in the log domain, also called the log likelihood ratio for each bit $x_{i,j}$ is approximated using the beliefs from GBP algorithm as

$$LLR(x_{i,j}) = \log \left(\frac{p(x_{i,j} = 1 | \mathbf{y})}{p(x_{i,j} = 0 | \mathbf{y})} \right) \approx \log \left(\frac{b(x_{i,j} = 1)}{b(x_{i,j} = 0)} \right).$$

The MAP detection minimizes the BER by maximizing the APP, $p(x_{i,j}|\mathbf{y})$, for each $x_{i,j}$ in \mathbf{x} .

In order to utilize the GBP algorithm for finding the LLRs, the first step is to identify the local constraint functions $f_{i,j}(\mathbf{x}_{i,j})$ given in (2.5). Since the 2-D ISI in our model is limited to a 3×3 span, the read-back sample $y_{i,j}$ and the corresponding media noise sample depends only on $\mathbf{x}_{i,j}$, the 3×3 bit region centered at (i, j) . Therefore, the local constraint functions can be defined using the pattern dependent noise distribution as $f_{i,j}(\mathbf{x}_{i,j}) = p(y_{i,j}|\mathbf{x}_{i,j})$.

We can incorporate the GBP algorithm to the probabilistic graphical model of this problem that we introduced in [29] in order to obtain the APPs. In order to obtain optimal performance, as seen in Section 2.2.3, we choose the regions to include all 3×3 regions and all possible intersections of these regions.

2.3 Using Soft Information from GBP for Iterative Decoding of Coded TDMR Channels

In our simulations, LDPC coded bits are written on and read from the Voronoi based magnetic medium, resulting in read-back samples. The GBP algorithm is used for signal detection and the LLRs from the GBP algorithm are used for iterative LDPC decoding using the belief propagation (BP) algorithm. We design the LDPC code rate by bounding the channel capacity of our TDMR channel model using the GBP based TDMR SIR estimation algorithm de-

Table 2.1: RS_{CT} (RS_{DT}) denotes the reader response span in cross-track (down-track) dimension. All the parameters in the table are specified in nanometers. \star indicates that the parameter is varied in the simulations. $CTC=7nm$.

| | TW | BP | RS_{CT} | RS_{DT} | PW_{50}^{CT} | PW_{50}^{DT} |
|------|---------|----|-----------|-----------|----------------|----------------|
| TDMR | \star | 10 | 28 | 28 | 14 | 14 |

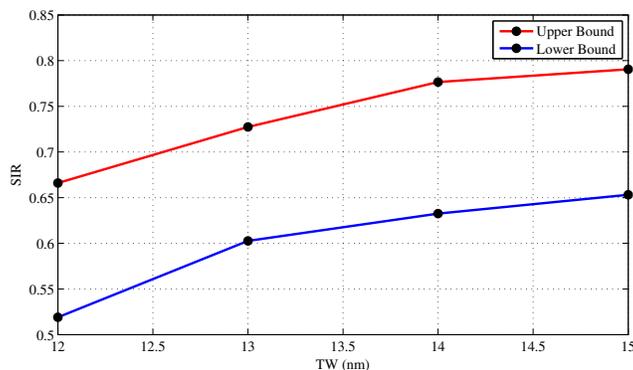


Figure 2.5: Lower and upper bounds on the SIR of Voronoi channel of TDMR system with the parameters given in Table 2.1

scribed in our recent work [36]. The TDMR channel model parameters used in the simulations are given in Table 2.1. The GBP algorithm detects a 32×32 page of data at a time. The read-head response is truncated to restrict the ISI span to 3×3 bit area.

2.4 Lower and Upper Bounds on the SIR of Voronoi Channel

In order to demonstrate the feasibility of implementation of our proposed GBP based TDMR detector, we conducted experiments to recover the user bits from the distorted coded TDMR channel.

The SIR between the input and output random processes X and Y of a Voronoi channel is defined as the mutual information per symbol between X and Y when the input distribution is uniform. For a $n \times m$ Voronoi channel, we have $SIR = \frac{1}{nm}I(X;Y)$ when the input distribution is uniform where $I(X;Y) = H(Y) - H(Y|X)$. The term, $H(Y|X)$, is the conditional entropy of

the media noise that can be computed analytically using the channel model. We estimated the media noise distribution $p(Y|\mathbf{x}_R)$ by an AWGN with the noise variance $\sigma_{\mathbf{x}_R}^2$ dependent on each 3×3 span of input data. Therefore, $H(Y|X = \mathbf{x}_R) = \frac{1}{2} \log(2\pi e \sigma_{\mathbf{x}_R}^2)$. $H(Y)$ is obtained using the GBP based TDMR SIR estimation algorithm [36].

The GBP-based capacity estimation algorithm provides a lower bound on the 2-D partition function of a factor graph, and accordingly the SIR which is obtained using the algorithm is only an estimate. In [24], GBP was used to estimate the capacity of 2-D RLL codes and it was shown that GBP capacity estimate for local constraints are accurate (up to 3rd decimal place). Moreover, in [37], it was shown that SIR, computed for the 2-D Gaussian channels using the GBP-based algorithm coincides with the lower and upper bounds of the SIR given by Chen and Siegel [38]. In our recent work [36], we have shown that the lower and upper bounds merge to the SIR of the Voronoi channel by increasing the dimensions of the medium. The upper and lower bounds on SIR are obtained as:

Lower Bound: No information about outside of boundaries of the 32×32 page is available for the GBP based TDMR SIR estimator. In this case, we compute the beliefs assuming that all states of the boundary regions are equiprobable. This gives us a lower bound on the SIR of the TDMR channel.

Upper Bound: The boundary information of the magnetic medium is assumed to be known to the SIR estimator. In this case, the bit values outside the page boundary are known and treated as deterministic giving us an upper bound on the SIR.

Figure 2.5 shows the SIR lower and upper bounds for the chosen TDMR channel model. We use SIR as a lower bound on the capacity of the system. Based on the observed lower and upper bounds of the SIR we choose LDPC

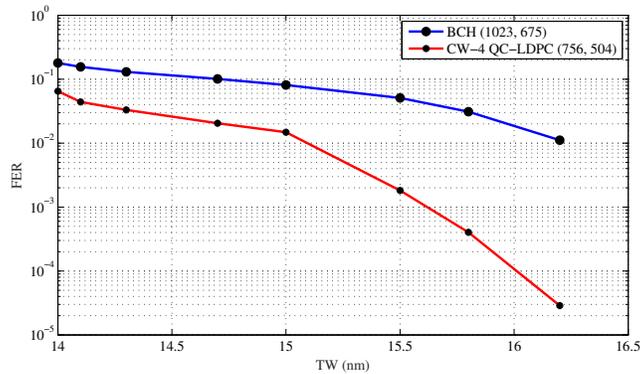


Figure 2.6: The FER result of quasi-cyclic column weight four LDPC code with $N = 756$, $R = 0.66$ with respect to the parameter TW for the Voronoi channel. Also, the FER of BCH code (1023, 675) is plotted for reference.

code rate to be $R = 0.66$. The LDPC code of length $N = 756$, rate $R = 0.66$, and a circulant size of $L = 126$ is constructed by methods described in [39], and is free of small trapping sets.

2.5 Frame Error Rate Results

Figure 2.6 shows the frame error rate (FER) results with respect to TW for the Voronoi channel with the parameters given in Table 2.1. At $TW = 16.2$ nm, the LDPC code gives more than two orders of magnitude gain in the FER when compared with the BCH code of length 1023 bits, rate 0.66.

CHAPTER 3

Investigation into Harmful Patterns over Two-Dimensional Magnetic
Recording

Many novel approaches have been recently proposed to increase the areal densities for magnetic recording systems beyond 1 Tb/in^2 . These technologies include heat assisted magnetic recording (HAMR) [40], bit patterned media (BPM) [41] and two dimensional magnetic recording (TDMR) [42]. TDMR is a purely systems driven approach centered around sophisticated signal processing and coding algorithms [43], [44] to achieve high areal densities; and can provide additive gains over HAMR and BPM technologies. In TDMR, the bits are densely packed leading to 2-D inter-symbol interference (ISI) and media noise that need to be mitigated via 2-D signal processing algorithms. Shingled magnetic recording (SMR) is a first step towards TDMR, where, the existing wide read/write heads are used to write tracks in an overlapping/shingled fashion. Since the TDMR technology is still emerging, several models for the TDMR channels at various interfaces are being proposed to facilitate the design of a viable read-channel architecture.

TDMR channel models for the media can be classified into a) discrete grain models, b) Voronoi media models and c) micro magnetic media models. Discrete grain models consider the recording medium as a tiling of grains of various known shapes on a 2-D plane. Voronoi models treat the distribution of grain centers as a point process. Micro magnetic models consider the sizes, shapes and distribution of the grains closely resembling the actual magnetic recording medium [31]. Recently, a communication theoretic framework [44]

was proposed to model TDMR channels by considering 2-D ISI from physical characteristics along with noise effects from the media and read electronics. In [44], though the jitter noise is modeled using a first order approximation and Gaussian statistics, the framework can be used to include the second order noise statistics empirically computed from the Voronoi model.

Efficient coding and signal processing algorithms are central for realizing areal density gains within TDMR systems. Several 2-D signal detection algorithms have been proposed over the last few years with an eye towards getting close to the maximum *a posteriori* (MAP)/maximum likelihood (ML) performance¹. Sullivan *et al.* [45] have proposed an iterative detection algorithm for 2-D ISI using 1D row-column detectors that iteratively exchange information to make soft-decision on the bit. A low complexity version of the algorithm optimized for separable 2-D ISI is proposed in [46]. Chen and Srinivasa [43] have proposed a 2-D joint equalization and detection (JTED) algorithm that combines a self iterating 2-D equalizer with multi-row-column detectors over the full signal span to iteratively achieve near MAP performance with tractable complexity. GBP algorithm is a different class of signal detection algorithms that uses message passing between regions instead of the message passing between nodes as seen in the traditional belief propagation algorithm. The performance of the GBP algorithm in relation to the MAP/ML algorithm is not known and requires a rigorous theoretical framework to study this. The GBP algorithm was studied by Khatami and Vasić [47] for different TDMR channel models.

Matcha et al. [48] have recently proposed a 2-D partial response ML for 2-D ISI channels using a 2-D soft-output Viterbi algorithm (SOVA) equivalent algorithm. The proposed method is within 1.5 dB of the full JTED perfor-

¹2-D MAP detection is NP hard.

mance with noise prediction [44]. While we have advanced methods for signal processing towards a full blown TDMR system, it is of practical interest to study shingled magnetic recording (SMR) systems using multi-track detection to assess areal density gains for read channels of immediate timely interest.

In this chapter, we demonstrate that how avoiding harmful patterns during the coding process leads to have a better detection performance in two dimensional magnetic recording (TDMR) systems. By avoiding such patterns at the source, we evaluate the performance of a multi-track detector and assess areal density gains over various TDMR system parameters. Furthermore, we explain the main idea of our method for generating 2-D constrained sequences achieving the capacity of constraint based on the GBP algorithm. Applied to a wide family of constraints, this method produces a convenient approach for investigating the benefits of implementing 2-D constrained waveforms in data storage systems.

This chapter is organized as follows. In Section 3.1, we describe the noise characteristics based on empirical results from the Voronoi media model and quantify the signal-to-noise ratio (SNR) using the peak power constraints. In Section 3.2, we describe a procedure for creating 2D constrained patterns satisfying the *no isolated bit* (n.i.b) constraint. We explain the main idea of our method for generating 2D constrained sequences achieving the 2D noiseless channel capacity for a wide family of constraints based on the GBP algorithm in Subsection 3.2.1. Finally, we evaluate the performance of various TDMR systems through simulations in Section 3.4.

3.1 Noise Characteristics of The TDMR Systems

In TDMR systems, the primary source of noise comes from irregular boundaries of grains and the random distribution of grain centers [31]. In addition,

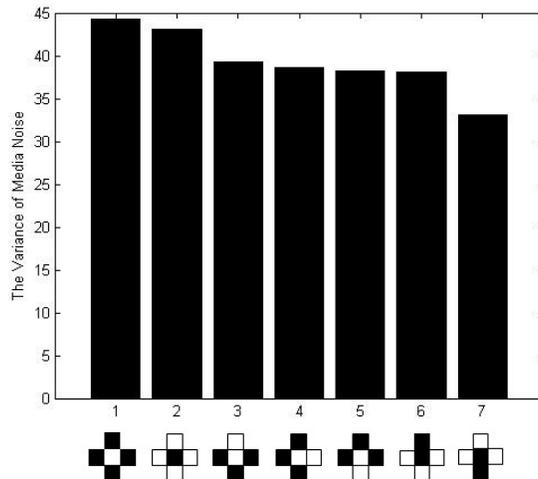


Figure 3.1: Observation of media noise variance for the Voronoi channel with the parameters $CTC = 7\text{nm}$, $BP = 7.5\text{nm}$ and $TW = 16\text{nm}$. In the 3×3 input patterns 0 and 1 are represented by white and black, respectively. It is shown that the harmful patterns for the Voronoi channel with 2-D ISI are ones eliminated by the no isolated bit constraint.

the noise distribution in TDMR is dependent on input information bits written on the Voronoi channel as the polarity of grains effects on the read-back signals. The Voronoi model with 2-D ISI [29] is considered as the magnetic recording channel. We first consider an ideal magnetic medium, where bit areas assumed to be rectangular, and then we apply the effect of irregular boundaries as the “media noise” [31] by an additive noise which is added to the read-back signal of ideal magnetic medium. We have analyzed the media noise characteristics for a read-head response of 2-D truncated Gaussian pulse with 3×3 span. Fig. 3.1 shows the media noise variance for different 3×3 input patterns. The media noise variance is greater for the input patterns with more transitions in cross-track and down-track direction. The most harmful input patterns are the ones with consecutive transitions in both cross-track and down-track directions. We have also observed the same characteristics of media noise studied with a more realistic channel model in [49]. In the sequel,

we introduce three definitions of SNR corresponding to overall noise, media noise and the electronic noise in TDMR systems.

Let $h_{i,j}(p, q)$ be the discrete-time response of $(i, j)^{\text{th}}$ bit. These response coefficients are random and dependent on the position and shape of grains within the bit area. The average bit-response is obtained by taking the expectation on these random response coefficients

$$h(p, q) = \mathbb{E}_{PQ} (h_{i,j}(p, q)), \quad (3.1)$$

where P and Q are random variables indicating the distribution of the grain positions in the down-track and cross-track directions, respectively. Therefore, the above averaging is taking into account all possible grain positions. The read-back signal sample without considering the electronic noise is given by

$$y_{i,j} = \sum_p \sum_q x_{i-p, j-q} h_{i-p, j-q}(p, q), \quad (3.2)$$

where $x_{i,j}$ is the symbol written on the $(i, j)^{\text{th}}$ bit-cell. Furthermore, the ideal read-head output, $s_{i,j}$, is obtained by considering the average discrete-time output of $(i, j)^{\text{th}}$ bit area as

$$s_{i,j} = \sum_p \sum_q x_{i-p, j-q} h(p, q). \quad (3.3)$$

The peak value of read-back signal, V_p , is defined by

$$V_p^2 = \sum_p \sum_q |h(p, q)|^2. \quad (3.4)$$

The media noise comes from the random perturbations of $h_{i,j}(p, q)$ around the average response $h(p, q)$. Therefore, the variance, or, equivalently the energy of media noise σ_m^2 is obtained by

$$\sigma_m^2 = \mathbb{E}_{P,Q} \left(\sum_p \sum_q |h_{i,j}(p, q) - h(p, q)|^2 \right). \quad (3.5)$$

Then, we can define three SNRs for a TDMR system according to the above definitions as

$$\begin{aligned} \text{SNR} &= 10 \log_{10} \left(\frac{V_p^2}{\sigma_m^2 + \sigma_e^2} \right), \\ \text{SNR}_{\text{Media}} &= 10 \log_{10} \left(\frac{V_p^2}{\sigma_m^2} \right), \\ \text{SNR}_{\text{Elec}} &= 10 \log_{10} \left(\frac{V_p^2}{\sigma_e^2} \right), \end{aligned} \quad (3.6)$$

where SNR is the overall SNR, and $\text{SNR}_{\text{Media}}$ and SNR_{Elec} are the SNRs corresponding to the media and electronic noise, respectively. A detailed description of these SNRs can be found in [44].

3.2 Evaluation of Utilizing Constrained Coded Data in TDMR systems

In this section, we investigate the performance gain due to using the constrained input waveforms in TDMR systems based on the BER criterion. In TDMR systems, decreasing the bit size to the limits comparable to the grain

size leads to a reduction in the SNR due to augmentation of the media noise. Since the media noise is caused by the polarity change in magnetization of neighboring grains due to consecutive transitions in the input data, low-pass constraints that restrict the consecutive transitions can be deployed to increase the SNR. Therefore, the constrained sequences can be deployed to reduce the harmful effects of the media noise. In addition to this, constrained coding reduces the state space of the detector and hence reduces the computational complexity of the detector.

3.2.1 Constrained Codes for Magnetic Recording Channels

The harmful data patterns contributing to high media noise are avoided using constrained codes. In our method, constraints are imposed locally and are given by a set of admissible input data patterns. Not all sequences of symbols from the input alphabet may be stored. Let $R_{\mathcal{C}}$ denote the rate of the code with a given constraint \mathcal{C} . To achieve the same storage density for a constrained coded system and an uncoded system, the rate loss due to the constrained input sequence is compensated by scaling the bit size of the coded system by a factor of $R_{\mathcal{C}}$. This reduction in bit size is justifiable only if the gain in performance due to constrained coding is high enough to compensate the effect of increased ISI. Therefore, the choice of the constrained code is dependent to the parameters of the TDMR system as well as the detector.

Let $\mathcal{S}_X \subset \{-1, +1\}^{N \times N}$ be a set of admissible $N \times N$ patterns for the constraint \mathcal{C} . An indicator function is defined as

$$f(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{S}_X, \\ 0, & \text{other,} \end{cases} \quad (3.7)$$

where \mathbf{x} is a random pattern. Consider a set of bit cells a in the neighborhood

of the cell (i, j) on the medium. Let x_a be a 2-D input pattern indexed by the elements of a , and $f_a(x_a)$ be the indicator function of x_a . f_a is referred to as a local constraint. As an example, the elements of a may correspond to the set of 3×3 bit cells with the center bit (i, j) . The indicator function of the $N \times N$ pattern is the product of all local constraints

$$f(\mathbf{x}) = \prod_a f_a(x_a). \quad (3.8)$$

Here, we introduce the *2-D no isolated bit* constraint which is utilized in the simulations of this chapter.

2-D No Isolated Bits (n.i.b.) Constraint: The input patterns which is a 1 surrounded by -1 's and a -1 surrounded by 1's are forbidden. This constraint is known as the no isolated bit constraint. The local constraint for the $(i, j)^{\text{th}}$ cell of the code is given as

$$f_a(x_{i-1,j}, x_{i+1,j}, x_{i,j}, x_{i,j-1}, x_{i,j+1}) = \begin{cases} 0, & x_{i-1,j} = x_{i+1,j} = x_{i,j-1} = x_{i,j+1} \neq x_{i,j}, \\ 1, & \text{other,} \end{cases} \quad (3.9)$$

where $x_{i,j}$ is the symbol written on the $(i, j)^{\text{th}}$ bit area of magnetic medium. In the following, we explain our method for generating 2-D constrained sequences achieving the 2-D noiseless capacity in the Appendix.

3.2.2 2-D Constrained Sequence Generator

In this subsection, we explain the main idea of using the GBP algorithm for generating 2-D constrained sequences achieving the maximum entropy of

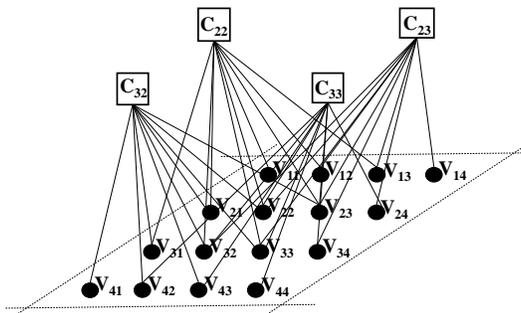


Figure 3.2: Factor graph of a 4×4 variable nodes with local constraints.

constraints. The GBP algorithm was utilized to estimate the 2-D noiseless capacity for a wide family of constraints in [29] and [24]. In order to obtain the capacity achieving distribution over the set of admissible patterns, the GBP as a capacity estimation algorithm is utilized. Then, we generate 2-D constrained sequences to write on a storage medium using the capacity achieving distribution. In order to utilize the GBP algorithm for generating 2-D constrained sequences, we need to introduce some preliminary definitions. We start introducing a graphical representation for the procedure as the GBP is a message passing algorithm.

- *The factor graph* corresponding to a local constraint is a bipartite graph consisting of a set of variable nodes $V_{i,j}$ (information bits) and a set of factor nodes $f_{C_{i,j}}$ (local constraints) in which a variable node $V_{i,j}$ is connected to a factor node $f_{C_{i,j}}$ if and only if $V_{i,j}$ is an argument of $f_{C_{i,j}}$. Fig. 3.2 shows an example of a factor graph for a 4×4 bit grid
- *The region graph* of the given graphical model is generated according to the cluster variation method [1]. In order to obtain the region graph, each parent region is specified by a set of variable nodes which are connected to the same factor node, i.e. for the set $C_{i,j}$ the parent region \mathcal{R}_i is equal to $\{V_{C_{i,j}}, f_{C_{i,j}}\}$, where $V_{C_{i,j}} = \{V_{i,j} | (i,j) \in C_{i,j}\}$. The other subregions are established by taking the intersection, the intersections of the

intersection, and so on of the parent regions. The region graph of the 4×4 cell square of variable nodes with 3×3 spans of the local constraints is established in Fig. 3.3.

- *Beliefs of each region \mathcal{R}_i* is the product of all the local factors in that region multiplied by all messages coming into region \mathcal{R}_i from outside region [1]. For each basic region \mathcal{R}_i , we have $2^{|\mathcal{R}_i|}$ beliefs of all possible cases for $|\mathcal{R}_i|$ variable nodes, in the binary domain, participated in the parent region which is denoted by $b_{\mathcal{R}_i}(x_{\mathcal{R}_i})$ where $x_{\mathcal{R}_i} \in \{-1, +1\}^{|\mathcal{R}_i|}$. The belief function is a good approximation of the marginal probability distribution of variables in a region.

The 2-D-noiseless channel capacity of a $N \times N$ array of 2-D constrained sequence is defined by

$$\mathcal{C}_{2-D} = \lim_{N \rightarrow \infty} \frac{\log_2(Z(N, N))}{N^2}, \quad (3.10)$$

where $Z(N, N)$, the 2-D partition function, specifies the number of legitimate patterns of the size $N \times N$ which satisfy the constraint. We can obtain the 2-D partition function by applying the GBP to the factor graph of a $N \times N$ variable nodes with local constraints. Since the Helmholtz free energy is $F_H = -\ln Z$, computing Z can be done by obtaining the region-based free energy estimate. If the GBP algorithm is used to estimate beliefs of each region $b(x_{\mathcal{R}_i})$ (or the marginal probability of each region), region-based free energy \hat{F}_H can be written as

$$\hat{F}_H = \sum_{\mathcal{R}_i \in \mathcal{R}} c_{\mathcal{R}_i} \sum_{x_{\mathcal{R}_i}} b_{\mathcal{R}_i}(x_{\mathcal{R}_i}) \left(\ln b_{\mathcal{R}_i}(x_{\mathcal{R}_i}) - \ln \prod_{a \in A_{\mathcal{R}}} f_a(x_a) \right), \quad (3.11)$$

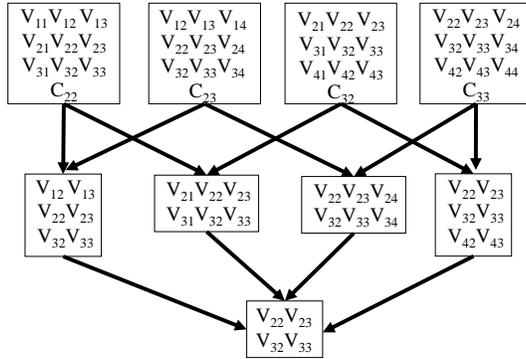


Figure 3.3: A region graph of a 4×4 variable nodes generated utilizing the parent to child scheme [1]

where \mathcal{R} is the set of all regions, c_{R_i} is the counting number defined as

$$c_{R_i} = 1 - \sum_{S \in \mathcal{S}_{R_i}} c_S, \quad (3.12)$$

where \mathcal{S}_{R_i} is the set of regions which are super-regions of R_i , x_{R_i} is the set of variables in R_i , and finally A_{R_i} is the set of local kernels in region R_i .

The main point is that the GBP as a capacity estimation algorithm provides the distribution over the admissible input patterns \mathbb{S} which achieves the 2-D noiseless channel capacity of constraint. According to the definition of 2-D partition function (the number of admissible patterns), we have

$$Z = \sum_{\mathbf{x} \in \mathbb{S}} f(\mathbf{x}), \quad (3.13)$$

where $f(\mathbf{x})$ is the indicator function. Then according to the above definition and the Z obtained from the GBP algorithm, we can write

$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{Z}. \quad (3.14)$$

where $p(x)$ is the distribution achieving the capacity of constraint. Therefore,

the probability distribution achieving the 2-D noiseless channel capacity with constraint coding is

$$p(\mathbf{x}) = \begin{cases} \frac{1}{|\mathbb{S}|}, & \mathbf{x} \in \mathcal{S}_X, \\ 0, & \text{other.} \end{cases} \quad (3.15)$$

Therefore, if we want to generate 2-D constrained sequences with maximum entropy, we need to obtain beliefs of regions (or the marginal probability of each region) which establishes the distribution $p(\mathbf{x})$ over the set of admissible patterns \mathbb{S} . Notice that the beliefs of forbidden patterns become 0, or, equivalently, the probability of occurrence of such patterns are 0. Then according to the obtained belief distribution achieving the 2-D noiseless channel capacity, 2-D constrained sequences are generated to write on a storage medium.

In this following, we provide a heuristic approach in order to generate constrained input with uniform distribution using the marginal probabilities estimated by the GBP algorithm. Inputs of algorithm are the given constraint \mathcal{C} , the region graph \mathcal{R} of a $N \times N$ variable nodes incorporated within local constraints, and the number of parent regions P of the region graph. We obtain the approximation of marginal probability distribution (beliefs) of the parent regions which achieve the constrained input with the uniform distribution for a given local constraint.

It should be noted that the beliefs of parent regions are previously computed and stored. Therefore, we define the steps of algorithm over the number of parent regions P . The first step in generating constrained sequences is assigning values to the variable nodes of first parent region using the beliefs of first parent region $b_{R_1}(x_{R_1})$, i.e., $X_1 \sim b_{R_1}(X_{R_1})$. At the i -th step, the values to the variables of i -th parent region are assigned. For this purpose, we define two sets of variable nodes for the parent region at step i :

- X_i^A is the set of variable nodes in the i -th parent region which were assigned in the previous steps.
- X_i^{NA} is the set of variable nodes in the i -th parent region which needed to be assigned at this step.

In addition, the contribution of variables which will be assigned in the next steps and are in the same parent region with the variables of set X_i^{NA} must be taken into consideration. We denote this set of variables with X_i^N . The distribution of X_i^{NA} to generate constrained sequences at the i -th step is given as

$$p(X_i^{NA}|X_i^A = x_i^A) = \sum_{x_i^N} p(X_i^{NA}, X_i^N = x_i^N | X_i^A = x_i^A). \quad (3.16)$$

These conditional distributions on the right hand side are obtained using the parent regions beliefs from GBP. The algorithmic description of GBP-based constrained sequence generator is given in Algorithm 1.

3.3 Detection Scheme

The read-back signal is detected using a multi-track MAP detector based on the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm. The BCJR algorithm provides the *a-posteriori* probability (APP) for each symbol given the detector input samples. The BCJR algorithm operates on the trellis representing the noiseless channel output sequences. It recursively computes the forward state metrics and the backward state metrics, which are combined with the branch metrics to produce the APP of each symbol. A detailed description of the BCJR algorithm can be found in [50].

In this study, we extend the BCJR algorithm to operate on the symbols

Algorithm 1: The GBP-Based Constrained Sequence Generation Algorithm

Input : \mathcal{R} the region graph,
 P the number of parent regions,
 \mathcal{C} the given constraint.

Output : \mathbf{x} the constrained sequence.

Initialization:

```

for  $i = 1$  to  $P$  do
    |
    |          $b_{R_i}(x_{R_i}) = \text{GBP}(\mathcal{R}, \mathcal{C});$ 
for  $p = 1$  to  $P$  do
    |
    |   if  $p = 1$  then
    |   |    $X_{R_1} \sim b_{R_1}(x_{R_1});$ 
    |   else
    |   |   foreach  $x_i^N$  do
    |   |   |    $X_i^{NA} \sim \sum_{x_i^N} p(X_i^{NA}, X_i^N = x_i^N | X_i^A = x_i^A);$ 
  
```

denoted by $x_{C_{i,j}} = \{x_{k,l} | (k,l) \in C_{i,j}\}$ instead of operating on the bit $x_{i,j}$. $x_{C_{i,j}}$ denotes the information bits contributing to the readback sample $y_{i,j}$, i.e., the bits at $C_{i,j}$ where $C_{i,j}$ denotes the 3×3 region with (i,j) as its center. In order to compute the bit error rate (BER) by using the BCJR algorithm, each trellis branch b at time k is assigned the metric

$$\mu(b_k) = p(y_{i,j}|b_k)p(b_k|x_{C_{i,j}}), \quad (3.17)$$

where $x_{C_{i,j}}$ is the starting (left-hand) 3×3 input state of b_k and $y_{i,j}$ is the output of the Voronoi channel corresponding to the input state $x_{C_{i,j}}$. In fact, $p(y_{i,j}|b_k)$ indicates the noise distribution of Voronoi channel.

As we assume the read-head response to be a 2-D truncated Gaussian pulse which spans 3×3 bit areas, the media noise is only dependent on a 3×3 span of input data. Based on extensive simulations, the media noise

distribution is shown to be close to the Gaussian distribution for most cases of the input states of a 3×3 bit region. Thus, we approximated the media noise distribution of each state of input $x_{C_{i,j}}$, i.e. each 3×3 bit region, with the Gaussian distribution with mean and variance dependent on input information. Therefore, we have

$$p(y_{i,j}|b_k) = \frac{1}{\sqrt{2\pi\sigma_{x_{C_{i,j}}}^2}} \exp\left(\frac{-(y_{i,j} - s_{i,j} - m_{x_{C_{i,j}}})^2}{2\sigma_{x_{C_{i,j}}}^2}\right), \quad (3.18)$$

where $m_{x_{C_{i,j}}}$ and $\sigma_{x_{C_{i,j}}}^2$ are the mean and variance of the media noise for the case of 3×3 input state $x_{C_{i,j}}$. For the case of ideal medium where the bit areas are in the form of rectangles, the discrete read-head output or “ideal values”, $s_{i,j}$, is obtained by convolving the magnetization pattern of the ideal recording medium with the read-head impulse response and sampling at the center of bit area in the down-track direction. The second term $p(b_k|x_{C_{i,j}})$ of the branch metric denotes the *a-priori* probability by which constrained sequences are generated. The *a-priori* probabilities for all the forbidden input patterns by the constraint are zero. The BER is obtained by applying the BCJR to the given trellis.

3.4 Evaluation of Performance of 2-D Constrained Codes in TDMR

We have simulated the TDMR system at different combinations of parameters denoted by TDMR(i), $1 \leq i \leq 4$, as given in the Table 3.1. The parameters chosen are realistic physical values and the parameter combinations TDMR(i) differ only in the size of each bit.

Fig. 3.4 compares the performances of the TDMR(1) and TDMR(2) configurations as a function of track-width in the absence of the electronic noise.

Table 3.1: RS_{CT} (RS_{DT}) denotes the reader response span in cross-track (down-track) dimension. CTC is assumed to be 7 nanometers. All the parameters in the table are specified in nanometers. \star indicates that the parameter is varied in the simulations.

| | TW | BP | RS_{CT} | RS_{DT} | TW_{50} | PW_{50} |
|---------|---------|-----|-----------|-----------|-----------|-----------|
| TDMR(1) | \star | 7.5 | 30 | 21 | 20 | 14 |
| TDMR(2) | \star | 7 | 30 | 21 | 20 | 14 |
| TDMR(3) | 16 | 7.5 | 30 | 21 | 20 | 14 |
| TDMR(4) | 16 | 7 | 30 | 21 | 20 | 14 |

In this comparison, the TDMR(1) configuration is used with unconstrained input while the TDMR(2) configuration is used with the n.i.b. constraint on the input sequences. To compensate for the rate loss due to the constrained coding, the BP in TDMR(2) in relation to the BP in TDMR(1) is chosen to match the rate of the n.i.b. constraint 0.9238, i.e.,

$$\frac{BP_{\text{TDMR}(2)}}{BP_{\text{TDMR}(1)}} \simeq 0.9238.$$

As it is shown in Fig. 3.4, using 2-D constrained sequences in TDMR systems improves the performance by about an order of magnitude. Not only the rate loss of constrained coding is compensated, but also an overall performance gain is obtained. Near the BER of 0.1, a 10% gain in the performance of a TDMR system is observed with storing only 2-D constrained sequences based on the BER criterion.

We add the electronic noise to the readhead's output of both TDMR(3) and TDMR(4) systems in order to find the SNR trade-off point where the 2-D n.i.b. constraint can compensate the effects of both media and electronic noises. The TDMR(3) is a constraint free system, but the TDMR(4)'s input sequences obey the n.i.b. constraint. Similar to the previous experiment, the BP of TDMR(4) is altered based on the rate of n.i.b. constraint. Let

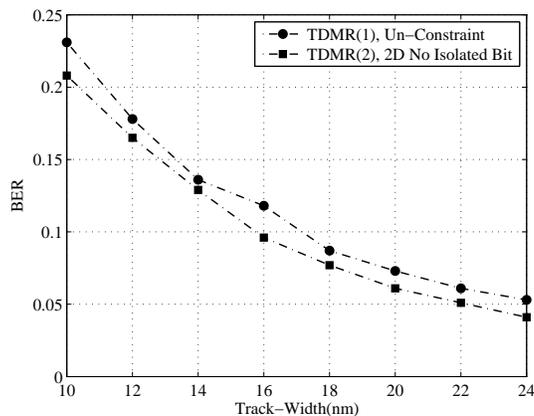


Figure 3.4: BER comparison of un-coded (TDMR(1)) and coded (TDMR(2)) systems with different bit areas and the same storage density in absence of electronic noise. Constrained coding improves the performance by avoiding the data patterns that result in high media noise

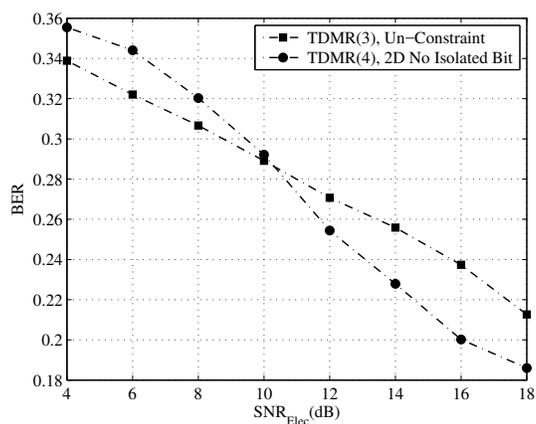


Figure 3.5: BER comparison of un-coded (TDMR(3)) and coded (TDMR(4)) systems with different bit areas and the same storage density in the presence of electronic noise. The impact of constrained coding is higher at high SNRs as the media noise dominates the electronic noise in this region. SNR_{Elec}=10 dB is a trade-off point where the performance gain due to constrained coding compensates the effects of both media and electronic noise.

σ_e^2 denotes the variance of electronic noise which is assumed to be Gaussian $\mathcal{N}(0, \sigma_e^2)$ and statistically independent of the media noise components in two-dimensions. The signal to noise ration corresponding to the electronic noise was defined in (3.6), where V_p^2 is the peak value of read-back signal. It can be seen from Fig. 3.5 that the $\text{SNR}_{\text{Elec}} = 10$ dB is the trade-off point between the performance gains of n.i.b. constrained coding and the effects of electronic and media noise. Constrained coding is targeted to handle the media noise, and hence is suitable to use at high SNRs where the media noise dominates the electronic noise. Therefore, at high SNRs, higher gains in BER performance is observed with the n.i.b. constraint giving an overall improvement over the TDMR(3) system.

CHAPTER 4

Constraint Gain for TDMR Channels

The constraint gain [51] is defined as the gap between the ultimate channel coding performance, in which a code is designed to satisfy both channel constraints and error correction code (ECC) constraints, and the average performance of the schemes where ECCs are designed separately without considering channel constraints. In TDMR systems, reducing track size for achieving higher areal densities results in significant signal-to-noise ratio (SNR) degradation and makes the media noise predominant [42]. The main source of media noise is transitions in the values of input bits written over neighboring bit cells in the magnetic medium (which comes from the irregularities of grains' boundaries over the magnetic medium). Two-Dimensional (2-D) transition limited constraints, which are typically low pass in nature, are imposed on input arrays in TDMR systems to mitigate the harmful effects of media noise. The benefits of using constrained codes come at the price of code rate penalty. However, this trade-off is a part of TDMR system design, balancing the operating SNR at a desired areal density point, as well as, facilitating reduced complexity signal detection by not allowing certain transitions in input data. Therefore, it is important to address the challenging problem of finding the trade-off between the rate loss of constrained codes and the ultimate performance gain of using them in TDMR systems. In principle, the ultimate coding approach for such data-dependent channels is to design a set of sufficiently spread codewords that also satisfy channel constraints [52, 53]. Furthermore, designing channel codewords satisfying both ECC and channel constraints is important as doing

this would achieve the noisy constrained-input channel capacity of channel and this is the maximum for the rate of any code for error-free transmission over a noisy constrained channel [51]. However, in practice this is difficult and we rely on sub-optimal methods such as forward concatenation method (standard concatenation), reverse concatenation method (modified concatenation) and combinations of these approaches [54–57].

In this chapter, we consider a Voronoi based channel model for TDMR systems as it gives a good trade-off between implementation complexity and the accuracy of modeling the media noise distribution. Furthermore, we consider a magnetic read-head which has a Gaussian sensitivity function that picks up magnetization from neighboring bit-cells over the magnetic medium. We investigate the performance gain of 2-D constraints using a lower bound estimate of the constraint gain for Voronoi based TDMR channels with realistic grain, bit, track and head dimensions. According to [51], a lower bound estimate on the constraint gain of a 2-D channel is the difference between the noisy max-entropic and uniform input capacities of the channel. We use schemes that employ the *Generalized Belief Propagation* algorithm for computing information rate estimates for TDMR channels [36, 58].

The chapter is organized as follows. In Section 4.1, we introduce the notion of constraint gain for 2D channels with memory. The GBP-based max-entropic information rate estimator is presented in Section 4.2. Furthermore, we investigate the accuracy of the GBP-based TDMR detector and information rate estimator in Section 4.3. Section 4.4 includes the simulation results of constraint gain for the 2D no-isolated-bit constraint over Voronoi based TDMR channels with different read-head and track dimensions.

4.1 Constraint Gain

In most of recording systems, some combinations of error correction and constrained codes are used to improve the performance. The design of joint error correction and constrained codes, i.e., designing the set of error correction codewords satisfying a constraint is a hard procedure. Mostly concatenations of linear block codes and constrained codes are used to impose both the error correction and channel constraints before recording on channels [54, 55]. For this, it is important to study and understand the performance gain of these methods in terms of capacity. In the following, we present the definition of a 2-D constraint and explain the notion of constraint gain, and uniform input and max-entropic information rates.

A 2-D binary constraint \mathcal{S}_C is the union $\bigcup_{m,n \in \mathbb{N}} \mathcal{S}_C^{m \times n}$ where $\mathcal{S}_C^{m \times n}$ denotes the set of all $m \times n$ arrays satisfying some predefined constraints. We can define the capacity of a 2-D constraint as follows

$$C_{2-D} = \lim_{m,n \rightarrow \infty} \frac{1}{m \times n} \log_2 Z(m, n), \quad (4.1)$$

where $Z(m, n)$ indicates the number of admissible $m \times n$ binary arrays.

A binary error correction encoder generates N -length codewords, \mathbf{c} , belonging to the set $\mathcal{S}_{ECC}^N \in \{0, 1\}^N$, where $N = m \times n$. The codewords are arranged into 2-D arrays of size $m \times n$. We are only interested in codewords satisfying the given 2-D constraint over $m \times n$ arrays, i.e., the codewords $\mathbf{c} \in \mathcal{S}_{ECC}^N \cap \mathcal{S}_C^{m \times n}$. The number of possible codewords satisfying both error correction and constrained code constraints is

$$\frac{1}{N} \log |\mathcal{S}_{ECC}^N \cap \mathcal{S}_C^{m \times n}|, \quad (4.2)$$

which is called the intersection rate in [51]. The intersection rate corresponds to the recording rate of N -length block code $\mathcal{S}_{\text{ECC}}^N$ which satisfy the constraint \mathcal{C} and is designed for a channel with a parameter θ . In [51], the rate of average intersection is defined as

$$R_{\text{avg ECC}}(\mathcal{C}, \theta) = \lim_{\epsilon \rightarrow 0} \lim_{m, n, N \rightarrow \infty} \frac{1}{N} \log \mathbb{E} \{ |\mathcal{S}_{\text{ECC}}^N \cap \mathcal{S}_{\mathcal{C}}^{m \times n}| \}, \quad (4.3)$$

where the expectation is taken over long enough (N, ϵ) codes and ϵ is defined as

$$\text{Cap}(\theta) - R_{\text{ECC}} \leq \epsilon. \quad (4.4)$$

Furthermore, $\text{Cap}(\theta)$ is the capacity of channel with the parameter θ

$$\text{Cap}(\theta) = \max_X I(X; Y), \quad (4.5)$$

in which maximum is taken over all stationary process X , Y is the corresponding output process and R_{ECC} is given by

$$R_{\text{ECC}} = \frac{1}{N} \log |\mathcal{S}_{\text{ECC}}^N|. \quad (4.6)$$

Furthermore, Fan *et al.* in [51] showed that this rate of average intersection can be obtained from

$$R_{\text{avg ECC}} = \text{Cap}(\theta) + C_{2\text{-D}} - 1, \quad (4.7)$$

where $\text{Cap}(\theta)$ is the noisy capacity of the channel with unconstrained inputs as given in Eq. (4.5), and $C_{2\text{-D}}$ is the noiseless channel capacity of constraint \mathcal{C} as given in Eq. (4.1). In fact, $R_{\text{avg ECC}}$ is the rate of average scheme over

all the schemes which jointly design ECC and constrained code codewords for the channel with the parameter θ . The lower bound on the rate of average intersection which is denoted by $R_{\text{lower ECC}}$ is given by

$$R_{\text{lower ECC}} = \max \{ \text{Cap}(\theta) + C_{2\text{-D}} - 1, 0 \}. \quad (4.8)$$

$R_{\text{lower ECC}}$ is the average rate of ECC (not necessarily linear codes) in which the ECC is designed without knowledge of constraint. Now, we need to find the maximum possible intersection rate to see how we can improve the lower bound of average intersection rate $R_{\text{lower ECC}}$ for a given channel.

We know the maximum achievable rate for a channel with constrained inputs is determined by the noisy constrained channel capacity as

$$\text{Cap}(\mathcal{C}, \theta) = \max_{X \in \mathcal{S}_{\mathcal{C}}} I(X; Y), \quad (4.9)$$

where the maximum is taken over all the stationary processes supported on the constraint $\mathcal{S}_{\mathcal{C}}^{m \times n}$. A process is supported on a set of constrained sequences if any finite sequence of strictly positive probability satisfies the constraint. The noisy constrained capacity

$$\text{Cap}(\mathcal{C}, \theta) \leq \min \{ \text{Cap}(\theta), C_{2\text{-D}} \}, \quad (4.10)$$

as it can not exceed the maximum entropy of input, or, the noiseless channel capacity of the constraint $C_{2\text{-D}}$, and the noisy constrained channel capacity can not be higher than the capacity of channel with unconstrained inputs as the maximum in (4.9) is taken only over the stationary processes which supported on $\mathcal{S}_{\mathcal{C}}^{m \times n}$. Similar to the average intersection rate, the maximum

rate intersection is defined as follows

$$R_{\max \text{ ECC}} = \lim_{\epsilon \rightarrow 0} \lim_{m, n, N \rightarrow \infty} \sup \frac{1}{N} \log \max \{ |\mathcal{S}_{\text{ECC}}^N \cap \mathcal{S}_{\mathcal{C}}^{m \times n}| \}, \quad (4.11)$$

where the maximum is taken over all possible (N, ϵ) good codes. Clearly, the maximum intersection rate can not be higher than the noisy constrained channel capacity, i.e.,

$$R_{\max \text{ ECC}} \leq \text{Cap}(\mathcal{C}, \theta). \quad (4.12)$$

The gap between the lower bound on the rate of average scheme and the noisy constrained channel capacity, or, equivalently, the upper bound on the maximum of intersection rate, is called the *Constraint Gain* for a channel with parameter θ and can be obtained from

$$\text{Constraint Gain}(\mathcal{C}, \theta) = |\text{Cap}(\mathcal{C}, \theta) - R_{\text{lower ECC}}|. \quad (4.13)$$

In fact, the Constraint Gain is the gap between the theoretical performance, in which the code is designed to satisfy the constrained code and ECC constraints and simultaneously this knowledge is exploited in the decoder, and the average performance of the schemes, where the ECC is designed separately without considering the constraint.

As it is well-known that computing the noisy constrained channel capacity is a hard problem for wide classes of channels, instead of computing the exact Constraint Gap, [51] proposed using the max-entropic capacity instead of $\text{Cap}(\mathcal{C}, \theta)$ in (4.13). Similar to (4.9), the max-entropic constrained capacity

for the channel can be defined as

$$\text{Cap}_{\max \text{ entropic}}(\mathcal{C}, \theta) = I(X_{\max}, Y_{\max}), \quad (4.14)$$

where X_{\max} is the input of channel which satisfies the constraint and is generated using the max-entropic distribution and Y_{\max} is the observation from the channel when input X_{\max} passing through the channel. The max entropic distribution can be obtained for constraints using message passing algorithms presented in [24, 37, 59]. By substituting $\text{Cap}_{\max}(\mathcal{C}, \theta)$ instead of $\text{Cap}(\mathcal{C}, \theta)$ in (4.13), we obtain an estimate of Constrained Gain as follows

$$\text{Constraint Gain}(\mathcal{C}, \theta) \simeq |\text{Cap}_{\max \text{ entropic}}(\mathcal{C}, \theta) - R_{\text{lower ECC}}|. \quad (4.15)$$

Here, we focus on methods providing an estimate of max-entropic information rate for Voronoi based TDMR channels using the GBP algorithm, as explained in the following.

4.2 Max-Entropic Information Rate

The max-entropic information rate of a Voronoi based TDMR channel with the pdf $p(\mathbf{y}|\mathbf{x})$ is defined as the mutual information rate between the max-entropic input and output as follows

$$\text{Cap}_{\max}(\mathcal{C}, \theta) = I(X_{\max}, Y_{\max}), \quad (4.16)$$

where X_{\max} is the input of channel which satisfies the constraint and is generated using the max-entropic distribution and Y_{\max} is the read-back samples from the Voronoi channel when input X_{\max} is written over the medium. Then,

we have

$$I(X_{\max}, Y_{\max}) = H(Y_{\max}) - H(Y_{\max}|X_{\max}), \quad (4.17)$$

where the input distribution is the max-entropic distribution, i.e., $p(\mathbf{x}) = \frac{1}{|\mathcal{S}_{\mathbf{C}}^{m \times n}|}$ and $|\cdot|$ indicates the cardinality.

The conditional entropy $H(Y_{\max}|X_{\max})$ can be obtained analytically using the media noise distribution $p(\mathbf{y}|\mathbf{x})$ and can be formulated as

$$H(Y_{\max}|X_{\max}) \stackrel{(a)}{=} \sum_{(i,j)} H(Y_{i,j}|X_{\max} = \mathbf{x}_{\mathbf{C}_{i,j}}) \stackrel{(b)}{=} \mathbb{E}_{X_{\max}} \frac{1}{2} \log \left(2\pi e \sigma_{\mathbf{x}_{\mathbf{C}_{i,j}}}^2 \right), \quad (4.18)$$

and

$$p(\mathbf{y}|\mathbf{x}) = \prod_{(i,j)} p(y_{i,j}|\mathbf{x}_{\mathbf{C}_{i,j}}), \quad (4.19)$$

$\mathbb{E}_{X_{\max}}$ is the expectation over all possible max-entropic inputs, and (b) is obtained as the pdf of Voronoi channel is a Gaussian distribution. Therefore, the problem of estimating the max-entropic information rate reduces to computing the entropy rate of the received output \mathbf{Y}_{\max} . For this purpose, we use the empirical averaging in the form of

$$H(\mathbf{Y}_{\max}) = -\mathbb{E}_{\mathbf{Y}_{\max}} \log p(\mathbf{y}) \approx -\frac{1}{L} \sum_{l=1}^L \log p(\mathbf{y}^{(l)}). \quad (4.20)$$

where L is the number of samples \mathbf{y} drawn according to $p(\mathbf{y})$. The constrained inputs are generated according to the distribution $p(\mathbf{x}) = \frac{1}{|\mathcal{S}_{\mathbf{C}}^{m \times n}|}$ and the pdf of channel is fixed for obtaining these L samples. Therefore, $p(\mathbf{y}^{(l)})$ can be

computed using

$$p(\mathbf{y}^{(l)}) = \sum_{\mathbf{x}} p(\mathbf{x})p(\mathbf{y}^{(l)}|\mathbf{x}) = \frac{1}{|\mathcal{S}_c^{m \times n}|} \sum_{\mathbf{x}} p(\mathbf{y}^{(l)}|\mathbf{x}), \quad (4.21)$$

where the right hand side equality is concluded by max-entropic distribution. Therefore, the problem of estimating the mutual information rate of a Voronoi based TDMR channel reduces to the problem of computing $\sum_{\mathbf{x}} p(\mathbf{y}^{(l)}|\mathbf{x})$, computing the marginal probabilities of a the probability distribution function $p(\mathbf{y}^{(l)}|\mathbf{x})$. We use techniques that incorporate the GBP algorithm for computing marginal probabilities of a probability distribution function [1, 24, 37] by finding an estimate of partition function of the factor graph representing the probability distribution function. We denote the partition function of the factor graph corresponding to $p(\mathbf{y}^{(l)}|\mathbf{x})$ by $Z(\mathbf{y}^{(l)})$, which is $Z(\mathbf{y}^{(l)}) = \sum_{\mathbf{x}} p(\mathbf{y}^{(l)}|\mathbf{x})$. We refer the reader to the original paper of GBP algorithm for further details [1]. The output entropy computation concludes to

$$\begin{aligned} H(Y_{\max}) &= -\frac{1}{L} \sum_{i=1}^L \log \left(\frac{1}{|\mathcal{S}_c^{m \times n}|} Z(\mathbf{y}^{(l)}) \right), \\ &= \log(|\mathcal{S}_c^{m \times n}|) - \frac{1}{L} \sum_{i=1}^L \log(Z(\mathbf{y}^{(l)})). \end{aligned} \quad (4.22)$$

4.3 On the Accuracy of The GBP-based TDMR Detector and Information Rate Estimator

The GBP algorithm provides a method to approximate marginal probabilities. The GBP algorithm is known to give exact marginals if and only if the region based graph has no loops [33]. In the sequel, we show that the GBP algorithm provides the marginals that are empirically close to the actual MAP marginals for our channel. In [24], GBP was used to estimate the capacity of certain 2D

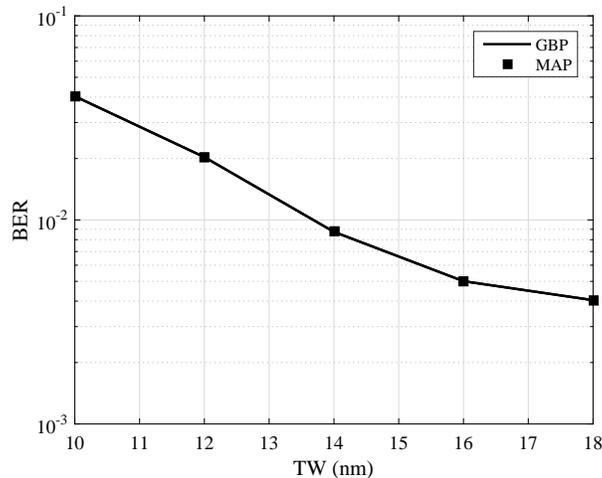


Figure 4.1: Hard-decision detection performance of a GBP detector versus optimal (MAP) detector error probability in terms of average BER per bits as a function of TW for a TDMR system. It should be noted that the GBP curve has no markers, but the MAP performance points, represented by markers alone, fall exactly on top of the GBP lines. The standard deviation of the results is small.

RLL codes and it was shown that GBP capacity estimate for local constraints are accurate (up to 3^{rd} decimal place). Moreover, in [37], it was shown that SIR, computed for the 2D Gaussian channels using the GBP-based algorithm coincides with the lower and upper bounds of the SIR given by Chen and Siegel [38].

Here, we evaluate the performance of the proposed GBP-based TDMR detector and information rate estimator. For this purpose, we simulate a 6×6 2D ISI Voronoi channel. The boundary information bits are assumed to have value (-1) . Fig. 4.1 compares the hard-decision detection performance of the optimal (MAP) detector and GBP-based TDMR detector in terms of average BER per bit as a function of TW . As can be seen, the GBP error decreases with TW and its performance is extremely close to the performance of MAP detection. Moreover, apart from providing the correct hard decisions, GBP infers the marginal probabilities. We observe empirically in all our exper-

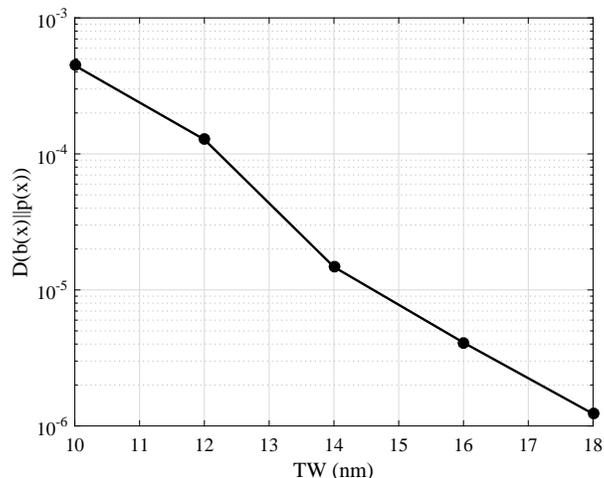


Figure 4.2: The KL-distance $D(b(\mathbf{x})||p(\mathbf{x}))$ between the beliefs $b(\mathbf{x})$ computed using GBP and marginals of optimal MAP $p(\mathbf{x})$ versus TW for a TDMR-based Voronoi channel.

iments that the marginal beliefs in GBP are accurately approximated. In order to investigate the accuracy of GBP-based TDMR information rate estimator, we show how well the marginal beliefs from the GBP approximate the MAP marginals using the KL-distance criterion. The KL-distance between two discrete distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}. \quad (4.23)$$

Fig. 4.2 shows the KL-distance between the marginal beliefs $b(\mathbf{x})$ inferred from GBP and the MAP marginals $p(\mathbf{x})$ for the 6×6 Voronoi channel of TDMR₉. As expected, based on the BER results shown in Fig. 4.1, the KL-distance between $b(\mathbf{x})$ and marginals of MAP is very small.

4.4 TDMR 2-D Constraint Gain Results

In this section, we present the max-entropic information rate and $R_{\text{lower ECC}}$ results for Voronoi based TDMR channels with different parameters. Similar

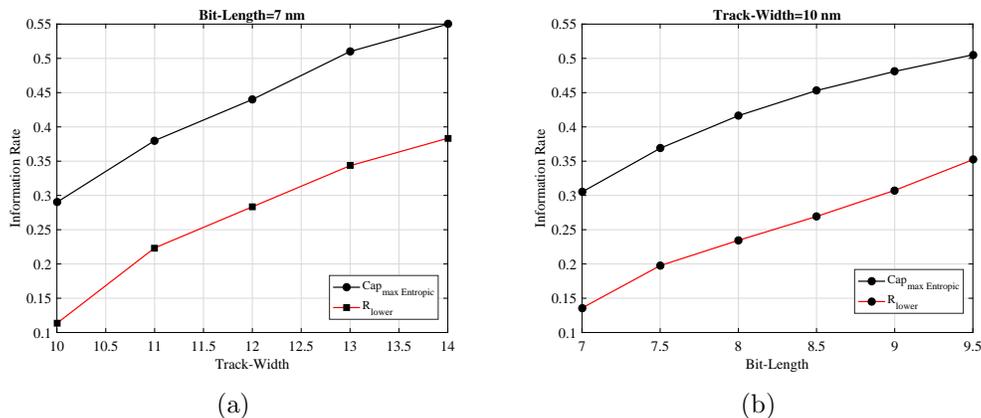


Figure 4.3: Estimating the constraint gain for the 2-D n.i.b. constraint over the Voronoi based TDMR channel with the parameters given in Table 4.1.

Table 4.1: RS_x (RS_y) denotes the reader response span in x -axis and y -axis directions, respectively. All the parameters in the table are in nanometers. \star indicates that the parameter varies in simulations.

| TW | BL | RS_x | RS_y | PW50 _x | PW50 _y |
|---------|---------|--------|--------|-------------------|-------------------|
| \star | 7 | 30 | 21 | 20 | 14 |
| 10 | \star | 30 | 21 | 20 | 14 |

to [58], we choose a read-head which has a Gaussian sensitivity function and spans 3×3 neighboring bit-cells over the magnetic medium. The parameters of the read-head and the Voronoi channel used in simulations are given in Table 3.1. The number of bit-cells in the x -axis and y -axis directions over the medium is 20×20 .

In Fig. 4.3, different information rate curves are given for the 2-D n.i.b. constraint over Voronoi based TDMR channels as functions of TW and BL. The R_{Lower} curve is obtained by the symmetric information rate curve shifted down by $1 - C_{2\text{-D n.i.b.}}$, where $C_{2\text{-D n.i.b.}} \simeq 0.9234$ is an estimate to the noiseless channel capacity of the 2-D n.i.b. constraint [60]. For some rate R , the constraint gain is estimated by the horizontal distance between the curves for $\text{Cap}_{\text{max Entropic}}$ and $R_{\text{Lower ECC}}$.

CHAPTER 5

Deliberate Bit Flipping Coding Scheme

Constrained codes have been used to overcome effects of harmful patterns in 1-D information storage systems. In [61], a systematic approach for designing 1-D constrained codes known as the state splitting algorithm is established. Marcus *et al.* used the results of the state splitting algorithm to design an encoder in the form of a finite state machine and a sliding window decoder with limited error propagation [62]. The theory of 1-D constrained coding is mature as well as practical aspects of 1-D code and decoder design. However, for the 2-D case it remains a challenge to design efficient, fixed-rate encoding and decoding algorithms (due to difficulty of certain problems that link to 2-D constraints compared to the 1-D case [63, 64]). A number of *variable-rate* encoding methods have been proposed for 2-D constrained channels, including bit-stuffing encoders [60, 65–67] and tiling based encoders [68, 69]. Furthermore, various *row-by-row* coding methods for specific 2-D constraints were presented in [70, 71]. Most of such 2-D constrained coding schemes have been proposed to achieve tighter bounds on the Shannon noiseless channel capacity of constraints. However, these schemes are non-linear, and their encoder/decoder has a memory such that over noisy channels single channel bit errors may cause a decoder to lose track of encoded bits and therefore propagate errors indefinitely without recovering.

In order to address the issue of error propagation in conventional constrained coding methods, Vasić and Pedagani proposed an alternative approach in [72], known as *deliberate bit flipping* (DBF), for applying binary

1-D $(0, k)$ -RLL constraint to error correction codewords (when k is large e.g., $k = 15$) to overcome the non-linear effects of 1-D constrained codes. Using a $(0, k)$ -RLL constraint monitor, a deliberate bit error is introduced into an error correction codeword whenever the number of consecutive zeros in the codeword reaches k . The method only relies on the capability of the ECC to correct both the deliberate errors and channel errors at the receiver. In [73–75], the problem of number of deliberate bit errors for imposing $(0, k)$ -RLL constraint into low-density parity-check (LDPC) codewords was partially addressed. Nevertheless, there is no attempt to minimize the number of bit flips for removing the forbidden configurations by the 1-D $(0, k)$ -RLL constraint from a given binary codeword. Moreover, the main problem with the DBF method introduced in [72] still is the number of deliberate bit errors that may overwhelm the ECC decoder and affect the error-floor performance (which limits its applications). Therefore, the key role of the DBF module should be to keep the number of flips small enough to not overburden the error correction decoder. The problem is also much more difficult for the 2-D case, and it is a challenge to design efficient algorithms for identifying harmful configurations in channel input patterns, let alone the problem of minimizing the number of bit flips.

In this chapter, we reformulate the problem of minimizing the number of bit flips in the DBF scheme for removing harmful configurations from 2-D channel input patterns as a constrained combinatorial optimization problem. Furthermore, we design a *(GBP)-guided DBF* algorithm for identifying 2-D harmful configurations and removing them with minimal number of flips. In order to use the GBP algorithm, we present a probabilistic graphical model for the constrained combinatorial minimization problem using the factor graph formulation in [1]. In this framework, patterns which do not contain harmful configurations are assumed to be uniformly distributed, and each pattern

containing a harmful configuration has zero probability. In this way, we reformulate the problem as a 2-D maximum *a posteriori* (MAP) problem, and demonstrate that the GBP algorithm can approximately solve this 2-D MAP problem. In order to study and analyze the performance of our proposed method, we introduce a binary 2-D channel with memory which captures the effect on an information bit from its surrounding patterns, i.e., the neighboring bits. These collections of adjoining bits are called *polyominoes*, objects studied in combinatorial mathematics [76]. The channel is viewed by a binary square tiling of a square lattice, where an information bit (0 or 1) is modeled by a white or black tile on the square lattice. The channel is characterized by rules defined by a set of configurations with a specific shape, which we call the set of *harmful configurations*. At the channel output, the probability of error for tiles contained in any of the harmful configurations are larger than for the other tiles. We evaluate the performance of the GBP-guided DBF method over the introduced channel where the 2-D isolated-bits configurations are considered as the channel harmful configurations. Furthermore, the performance of the DBF method for 2-D *no isolated-bits* (n.i.b.) constraint on a memoryless binary symmetric channel (BSC) is compared with the row-by-row and bit-stuffing based 2-D n.i.b. encoders, presented in [60] and [77], respectively.

The rest of this chapter is organized as follows. Section 5.1 presents the notations and definitions used throughout the paper. In Section 5.2, the data-dependent channel model is introduced. In Section 5.3, the problem of minimizing the number of flipped bits in the DBF method is formulated. In Section 5.4, we reformulate the minimization problem as a 2-D MAP problem, and explain the ideas of using the GBP algorithm for solving this problem. Numerical results are presented in Section 5.5.

5.1 Tilings and Polyominoes

We denote a discrete random variable with an upper case letter (e.g., X) and its realization by the lower case letter (e.g., x). We denote the probability density function of X with $p(x)$ and the conditional probability density function of Y given X by $p(y|x)$. $[n_1 : k : n_2]$ represents the set of real numbers $\{n_1, n_1 + k, n_1 + 2k \dots, n_2\}$, and $[n]$ denotes $[1 : 1 : n]$. We denote a random array of size $m \times n$ by $\mathbf{X} = [X_{i,j}]_{i \in [m], j \in [n]}$. An array of binary symbols with size $m \times n$ is denoted by $\mathbf{x} = [x_{i,j}]_{i \in [m], j \in [n]}$ where $x_{i,j} \in \{0, 1\}$ is the $(i, j)^{\text{th}}$ component of array. $\mathcal{A}_{m,n} = \{(i, j) \in \mathbb{Z}^2 : i \in [m] \text{ and } j \in [n]\}$ denotes the index set of an array of size $m \times n$ and is the subset of the 2-D lattice \mathbb{Z}^2 . The Hamming weight of an array \mathbf{x} of binary symbols is determined by

$$w_H(\mathbf{x}) = \sum_{x_{i,j} \in \mathbf{x}} \mathbb{1}\{x_{i,j} = 1\}, \quad (5.1)$$

where $\mathbb{1}\{\cdot\}$ equals one (respectively, zero) when its argument is true (respectively, false). The XOR operation between two binary arrays (\mathbf{x} and \mathbf{y} of size $m \times n$) is done component-wise, i.e., $\mathbf{x} \oplus \mathbf{y} = (z_{i,j})_{i \in [m], j \in [n]}$ where $z_{i,j} = x_{i,j} \oplus y_{i,j}$, and $x_{i,j}$ and $y_{i,j}$ are the $(i, j)^{\text{th}}$ component of \mathbf{x} and \mathbf{y} , respectively. Furthermore, the Hamming distance between \mathbf{x} and \mathbf{y} is determined by $d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} \oplus \mathbf{y})$. A binary BCH code of length N with $N - K$ parity bits and minimum distance d_{\min} is denoted by BCH- $[N, K, d_{\min}]$.

A *tiling* of the plane is a collection of plane figures that fills the plane with no overlaps and no gaps. The plane figures used as building blocks for tilings are called *tiles*. A *polyomino* of order k , called also a *k-ominoe*, is a plane geometric figure formed by joining k neighboring square tiles. Among



Figure 5.1: Two examples of polyominoes: (a) a 2×2 square and (b) a cross.

polyominoes are 2×2 square-shaped polyominoes

$$Q^\square(i, j) = \{(i, j), (i, j + 1), (i + 1, j), (i + 1, j + 1)\}, \quad (5.2)$$

and cross-shaped polyominoes

$$Q^+(i, j) = \{(i, j - 1), (i - 1, j), (i, j), (i, j + 1), (i + 1, j)\}, \quad (5.3)$$

over the 2-D lattice \mathbb{Z}^2 , which are shown in Fig. 5.1.

A tiling is said to be *colored* or *labeled* if each of its tiles is assigned a color/symbol from a finite set of colors/symbols. A binary coloring or labeling employs *black* and *white* tiles. A colored tiling is also referred as a *pattern* or a *configuration*. A square binary tiling of an $m \times n$ rectangle ($m \times n$ binary pattern) is denoted by $\mathbf{x} = [x_{i,j}]_{i \in [m], j \in [n]}$, where $x_{i,j}$ indicates the color of tile in i -th row and j -th column, and $x_{i,j} = 0$ represents a white tile and $x_{i,j} = 1$ a black tile. Consider a k -ominoe \mathcal{P} and the set of all 2^k binary configurations of that shape $\mathcal{X}_{\mathcal{P}}$. We refer to them as to \mathcal{P} -shaped configurations and denote them by $\mathbf{x}_{\mathcal{P}}$.

Consider the tile (i, j) over an $m \times n$ rectangular pattern \mathbf{x} , then the union of all \mathcal{P} -shaped polyominoes that intersect with this tile is denoted by $\mathcal{P}_{i,j}$. The configuration of $\mathcal{P}_{i,j}$ is denoted by $\mathbf{x}_{\mathcal{P}_{i,j}}$. For the cases of 2×2 square-shaped and cross-shaped polyominoes, we have

$$\mathcal{P}_{i,j}^{\square} = \bigcup_{(i',j') \in Q^{\square}(i-1,j-1)} Q^{\square}(i',j'), \quad (5.4)$$

and

$$\mathcal{P}_{i,j}^{+} = \bigcup_{(i',j') \in Q^{+}(i,j)} Q^{+}(i',j'), \quad (5.5)$$

respectively. Fig. 5.2 shows $\mathcal{P}_{i,j}$ for these polyominoes.

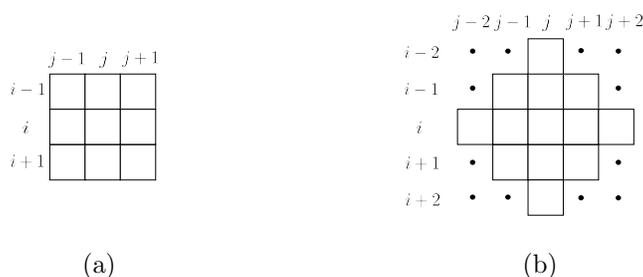


Figure 5.2: Figure demonstrates $\mathcal{P}_{i,j}$ over a rectangle when the polyomino is: (a) a 2×2 square and (b) a cross.

5.2 Channel Model

In this section, we introduce a communication channel transmitting binary rectangular patterns and producing as an output a binary pattern. Passing through the channel, a tile is in error if its color gets inverted. The channel is data-dependent and characterized by rules defined by a set of binary configurations of a \mathcal{P} -shaped polyomino. We call this set of \mathcal{P} -shaped configurations the set of *harmful configurations*. At the channel output, the error probability of binary tiles contained in configurations which belong to the set of harmful configurations is larger than the other tiles. Therefore, the channel has states and its error statistics depends on input binary patterns. In the following, we formally present error and state characterizations.

The input and output alphabets \mathcal{X} and \mathcal{Y} are two sets of binary rectangular

patterns of size $m \times n$. An $m \times n$ binary pattern $\mathbf{x} = [x_{i,j}]_{i \in [m], j \in [n]}$ is chosen randomly and uniformly from \mathcal{X} as an input to the channel. The channel output, $\mathbf{y} = [y_{i,j}]_{i \in [m], j \in [n]} \in \mathcal{Y}$, is also a binary pattern of size $m \times n$. For the tile (i, j) , $\mathcal{P}_{i,j}$ denotes the union of \mathcal{P} -shaped polyominoes that intersect with this tile, and $\mathbf{x}_{\mathcal{P}_{i,j}}$ is the configuration of $\mathcal{P}_{i,j}$, as defined in Section 5.1. We assume that the set of all possible configurations for $\mathcal{P}_{i,j}$, denoted by $\mathcal{X}_{\mathcal{P}_{i,j}}$, can be partitioned into two disjoint subsets $\mathcal{X}_{\mathcal{P}_{i,j}}^G$ and $\mathcal{X}_{\mathcal{P}_{i,j}}^B$, i.e., $\mathcal{X}_{\mathcal{P}_{i,j}} = \mathcal{X}_{\mathcal{P}_{i,j}}^G \cup \mathcal{X}_{\mathcal{P}_{i,j}}^B$, where $\mathcal{X}_{\mathcal{P}_{i,j}}^B$ is the set of configurations containing \mathcal{P} -shaped configurations which are harmful for the channel. For example, $\mathcal{X}_{\mathcal{P}_{i,j}}^B$ can be the set of binary configurations of $\mathcal{P}_{i,j}$ given in Fig. 5.2(b), which contains the 2-D *isolated-bit* patterns.

For a binary tile $x_{i,j}$ contained in a harmful \mathcal{P} -shaped configuration, the channel is in the bad state, and the probability of error is α_b . However, passing through the channel, a binary tile that does not belong to a harmful configuration is in error with a probability of α_g , and the channel is in the good state. We assume that $\alpha_b \gg \alpha_g$, or, in other words, the probability of error for tiles contained in a harmful configuration is much larger than that of the other tiles. The received binary pattern is $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}^{\text{CH}}$, where $\mathbf{e}^{\text{CH}} = [e_{i,j}^{\text{CH}}]$ is the channel error array and denotes the locations of tiles whose colors are inverted passing through the channel. Therefore, $e_{i,j}^{\text{CH}}$ has either Bernoulli(α_g) or Bernoulli(α_b) distribution, depending on the pattern $\mathbf{x}_{\mathcal{P}_{i,j}}$. In fact, the channel is a binary symmetric channel (BSC) with crossover probability α_b when $\mathbf{x}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B$ and a BSC with crossover probability α_g when $\mathbf{x}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}_{i,j}}^B$, respectively.

We define an *indicator* function for the channel $f_{\text{CH}} : \mathcal{X}_{\mathcal{P}_{i,j}} \rightarrow \{0, 1\}$ over every tile (i, j) ,

$$f_{\text{CH}}(\mathbf{x}_{\mathcal{P}_{i,j}}) = \mathbb{1} \left\{ \mathbf{x}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B \right\}, \quad (5.6)$$

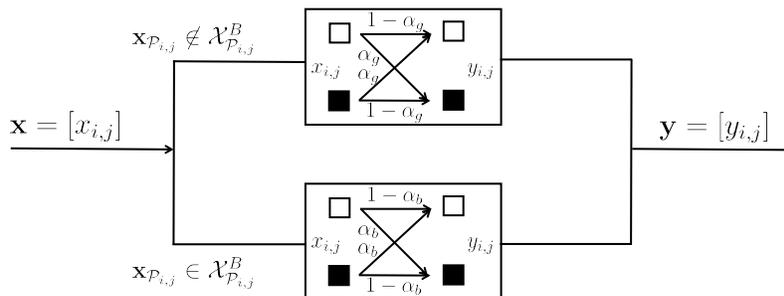


Figure 5.3: A schematic representation for the channel model is given. Passing through the channel, the color of tile $x_{i,j}$ inverts with probability α_b if the configuration of $\mathcal{P}_{i,j}$, $\mathbf{x}_{\mathcal{P}_{i,j}}$, belongs to the set of harmful patterns $\mathcal{X}_{\mathcal{P}_{i,j}}^B$, otherwise it inverts with a probability of α_g .

to identify tiles which are contained in harmful configurations, where the tile (i, j) belongs to at least one harmful configuration if $f_{\text{CH}}(\mathbf{x}_{\mathcal{P}_{i,j}}) = 1$. Using the above indicator function, we can determine the channel state for transmission of tile (i, j) as follows where “ b ” and “ g ” stand for the bad and the good channel states, respectively. Let the probability distribution function of channel to be $p(\mathbf{y}|\mathbf{x})$. According to the aforementioned error characterization, the probability distribution function of channel can be factored into

$$p(\mathbf{y} | \mathbf{x}) = \prod_{(i,j)} p(y_{i,j} | \mathbf{x}_{\mathcal{P}_{i,j}}), \quad (5.7)$$

since the configuration of output tile $y_{i,j}$ only depends on the configuration of $\mathcal{P}_{i,j}$ in the input pattern \mathbf{x} . Fig. 5.3 gives a schematic illustration for the channel.

Remark 1 *The theory of domino tilings of lattices are widely used in data storage applications for capacity estimation and constrained coding, some notable examples are [69, 78–80]. This is due to the fact that the theory of domino tilings is well studied [81–84]. In this paper, we only focus on 4-ominoes and*

5-ominoes, as these reflect physical effects of 2-D ISI over the plane. For this purpose, we defined the square and cross shaped polyominoes in (5.2) and (5.3).

Remark 2 *The channel is similar to the Gilbert-Elliot channel [85], as it has two states, where in each state acts as a BSC with a different cross-over probability. However, the state transitions in our channel model depend on input patterns. For such channels, calculating the information rate, let alone the capacity, is much more challenging than for discrete memoryless channels. Except for very special cases, there are no simple expressions for information rates available, and so, one needs to rely on upper and lower bounds and/or on stochastic techniques for estimating the information rate, examples are [86–88].*

Remark 3 *The probability that the channel is in the bad state (or, in the good state) depends on the input probability distribution. If we assume that input bits are i.i.d., then there is no Markovian assumption on the channel states. The probability that the channel is in the bad state for sending the tile (i, j) is*

$$p(s_{i,j} = b) = p(f_{CH}(\mathbf{x}_{\mathcal{P}_{i,j}}) = 1) = \frac{|\mathcal{X}_{\mathcal{P}_{i,j}}^B|}{|\mathcal{X}_{\mathcal{P}_{i,j}}|}, \quad (5.8)$$

as the patterns are chosen randomly and uniformly, and in the good state is $p(s_{i,j} = g) = 1 - p(s_{i,j} = b)$. For different input probability distributions, this probability can be computed accordingly. Throughout the paper, we do not consider any Markovian properties on input tiles.

In the following, we present an example of an input binary pattern to the channel, where the 2-D isolated-bits patterns are the harmful patterns for the channel, to illustrate the effects of harmful patterns on input tiles passing through the channel.

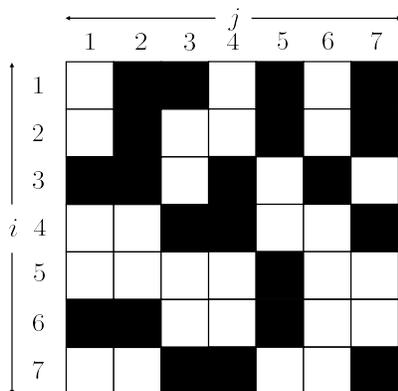


Figure 5.4: A 7×7 binary pattern \mathbf{x} is transmitted through the channel with the set of 2-D isolated-bits patterns as the set of harmful patterns. The tiles $(2,6)$, $(3,5)$, $(3,6)$, $(3,7)$, $(4,6)$, $(6,7)$, $(7,6)$ and $(7,7)$ belong to the 2-D isolated-bits patterns. Passing through the channel, the probability of error for these tiles is α_b , and for the rest of tiles is α_g .

Example 1 Fig. 5.4 shows an example of a 7×7 input binary pattern \mathbf{x} transmitted over the introduced channel. We assume that the set of harmful patterns for the channel is the set of 2-D isolated-bits patterns. In order to determine the channel state for all tiles over the pattern, we assume zero entries (white tiles) outside of \mathbf{x} , i.e., $x_{i,j} = 0$, while $i < 1$, $j < 1$, $i > 7$, or $j > 7$. There are two isolated-bits patterns in \mathbf{x} , which are $\mathbf{x}_{Q^+(3,6)}$ and $\mathbf{x}_{Q^+(7,7)}$. Passing through the channel, the tiles contained in these two harmful configurations are in error with a probability of α_b . These tiles are $(2,6)$, $(3,5)$, $(3,6)$, $(3,7)$, $(4,6)$, $(6,7)$, $(7,6)$ and $(7,7)$. For instance, for the tile $(2,6)$,

$$\mathcal{P}_{2,6} = \bigcup_{(i',j') \in Q^+(2,6)} Q^+(i',j'). \quad (5.9)$$

Since $Q^+(3,6) \subset \mathcal{P}_{2,6}$ and $\mathbf{x}_{Q^+(3,6)}$ is a 2-D isolated-bits pattern, we have the fact that $\mathbf{x}_{\mathcal{P}_{2,6}}$ contains a 2-D isolated-bits pattern, and therefore, the tile $(2,6)$ is in the bad state. Similarly, we can check this for the rest of tiles in \mathbf{x} .

5.3 Problem Formulation

The user uniformly and randomly selects a binary message \mathbf{m} out of 2^K messages denoted by $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{2^K}\}$, where each message is of length $K \in \mathbb{N}$. The user message \mathbf{m} is first encoded by an error correction encoder with rate $R = \frac{K}{N}$. The *error correction encoding* function $\phi_{\text{ECC}} : \mathcal{M} \rightarrow \mathcal{S}_{\text{ECC}}^N$ assigns a binary codeword $\mathbf{c}(\mathbf{m})$ of length N to the user data \mathbf{m} such that

$$\mathbf{c}(\mathbf{m}) = \phi_{\text{ECC}}(\mathbf{m}), \quad (5.10)$$

where $\mathcal{S}_{\text{ECC}}^N = \{\mathbf{c}(\mathbf{m}_1), \mathbf{c}(\mathbf{m}_2), \dots, \mathbf{c}(\mathbf{m}_{2^{\lfloor NR \rfloor})}\}$ is the codebook (the set of binary codewords of length N) associated with the ECC being used. A codeword $\mathbf{c} \in \mathcal{S}_{\text{ECC}}^N$ is represented by N binary symbols, $\mathbf{c} = (c_1, c_2, \dots, c_N)$, and $N = m \times n$. Each codeword is arranged into an array \mathbf{x} of size $m \times n$, such that $\mathbf{x} = [x_{i,j}]_{i \in [m], j \in [n]}$, and $x_{i,j} = c_{(i-1)m+j}$. The array \mathbf{x} can be considered as a binary rectangular pattern of size $m \times n$. We want to send the pattern \mathbf{x} over the communication channel in Section 5.2, with the list of harmful configurations $\mathcal{X}_{\mathcal{P}_{i,j}}^B$. Assuming that $\alpha_b \gg \alpha_g$, then binary tiles contained in configurations of list $\mathcal{X}_{\mathcal{P}_{i,j}}^B$ are more prone to error than the other tiles. To overcome effects of harmful configurations, we use a deliberate error insertion approach to remove the harmful configurations from the input pattern \mathbf{x} before transmission through the channel. Whenever there is a configuration from the list $\mathcal{X}_{\mathcal{P}_{i,j}}^B$ in the input pattern \mathbf{x} , the color of selected tiles in \mathbf{x} are inverted to remove the harmful configurations. We denote the set of $m \times n$ binary patterns which do not contain the harmful configurations by \mathbb{S} . For the 7×7 pattern \mathbf{x} in Example 1, we can remove the 2-D isolated-bits patterns from the given 7×7 binary pattern by inverting the colors of tiles (3, 6) and (7, 7). This method of eliminating harmful configurations from binary patterns with

inverting the color of tiles can be viewed as the mapping ϕ from the set of $m \times n$ binary patterns \mathcal{X} to a set of $m \times n$ binary patterns \mathbb{S} that do not contain the harmful configurations. The mapping function $\phi : \mathcal{X} \rightarrow \mathbb{S}$ assigns an $m \times n$ binary pattern $\hat{\mathbf{x}}$ to the input pattern \mathbf{x} so that

$$\hat{\mathbf{x}} = \phi(\mathbf{x}). \quad (5.11)$$

Let $\theta : \mathcal{X} \rightarrow \{0, 1\}^{m \times n}$ be the function selecting tiles whose colors need to be inverted for removing the harmful configurations from the pattern \mathbf{x} . Using the function θ , we define \mathbf{e}^{DBF} to identify the positions of tiles whose colors are inverted,

$$\mathbf{e}^{\text{DBF}} = \theta(\mathbf{x}) = [e_{i,j}^{\text{DBF}}]_{i \in [m], j \in [n]}, \quad (5.12)$$

where $e_{i,j}^{\text{DBF}} = 1$ if the color of (i, j) -th tile is inverted, otherwise, $e_{i,j}^{\text{DBF}} = 0$. Therefore, $\mathbf{x} \oplus \mathbf{e}^{\text{DBF}}$ does not contain any \mathcal{P} -shaped harmful configurations from the list $\mathcal{X}_{\mathcal{P}_{i,j}}^{\mathcal{B}}$. Furthermore, we have

$$\phi(\mathbf{x}) = \mathbf{x} \oplus \theta(\mathbf{x}), \quad (5.13)$$

and the number of tiles whose colors are inverted is equal to $w_H(\mathbf{e}^{\text{DBF}})$. Now, $\hat{\mathbf{x}}$ is transmitted over the channel instead of \mathbf{x} , and the $m \times n$ binary pattern \mathbf{y} is received. We identify the locations of channel errors by the array \mathbf{e}_{CH} which is $\hat{\mathbf{x}} \oplus \mathbf{y}$. Then, if the chosen message is \mathbf{m} , since $\mathbf{y} = \hat{\mathbf{x}} \oplus \mathbf{e}_{\text{CH}}$ and $\hat{\mathbf{x}} = \mathbf{x}(\mathbf{m}) \oplus \mathbf{e}^{\text{DBF}}$, we have

$$\mathbf{y} = \mathbf{x} \oplus \mathbf{e}^{\text{CH}} \oplus \mathbf{e}^{\text{DBF}}. \quad (5.14)$$

Naturally, such an encoder will have a corresponding decoder (let us denote the decoder by ψ). The decoder ψ assigns an estimate of $\hat{\mathbf{m}} \in \mathcal{M}$ to each

received pattern \mathbf{y} from the channel such that

$$\psi : \mathcal{Y} \rightarrow \mathcal{M}, \quad (5.15)$$

where $\hat{\mathbf{m}} = \psi(\mathbf{y})$. The performance of this deliberate error insertion method is measured by the probability that the estimate of the message $\hat{\mathbf{m}}$ is different from the actual message \mathbf{m} . Let $\lambda_{\mathbf{m}} = p(\hat{\mathbf{m}} \neq \mathbf{m} | \mathbf{m})$ be the probability of error given that the actual message is \mathbf{m} . Then, the average probability of error is given by

$$p_e^{(N)} = p(\hat{\mathbf{m}} \neq \mathbf{m}) = \sum_{\mathbf{m} \in \mathcal{M}} \lambda_{\mathbf{m}} p(\mathbf{m}) \stackrel{(a)}{=} \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} \lambda_{\mathbf{m}}, \quad (5.16)$$

where (a) comes from the fact that \mathbf{m} is chosen uniformly from the set \mathcal{M} and $|\mathcal{M}| = \frac{1}{2^{\lfloor NR \rfloor}}$. A rate R is said to be achievable if, given an $\epsilon > 0$, there exists an N_ϵ such that $p_e^{(N_\epsilon)} \leq \epsilon$. The capacity of the method is defined as the supremum over all achievable rates.

We assume that the decoder ψ is a bounded-distance decoder which should ideally be able to retrieve the binary user data from the received pattern \mathbf{y} for every message $\mathbf{m} \in \mathcal{M}$. This bounded-distance decoder can correct the error patterns with Hamming weights lying within the error correction capability of the code, i.e., if

$$d_H(\mathbf{x}(\mathbf{m}), \mathbf{y}) \leq \lfloor \frac{d_{\min} - 1}{2} \rfloor, \quad (5.17)$$

where d_{\min} is the minimum distance of the code, the decoder should be able to correct the errors. There are two types of errors in this communication system with the deliberate error insertion method. The first type is the deliberate errors for removing harmful configurations from the input pattern. The second is the channel errors which may have or may not have overlaps with the

deliberate errors. Since appearances of harmful patterns in the input pattern dominate the channel errors, we can assume that $w_H(\mathbf{e}^{\text{CH}}) \simeq 0$ after removing harmful patterns from the input pattern. Under this assumption, we have $\mathbf{y} \simeq \mathbf{x} \oplus \mathbf{e}^{\text{DBF}}$ and

$$d_H(\mathbf{x}, \mathbf{y}) \simeq d_H(\mathbf{x}, \mathbf{x} \oplus \mathbf{e}^{\text{DBF}}) = w_H(\mathbf{e}^{\text{DBF}}). \quad (5.18)$$

Therefore, if $w_H(\mathbf{e}^{\text{DBF}}) \leq \lfloor \frac{d_{\min}-1}{2} \rfloor$, the decoder can correct the errors. For this case, the probability of error for retrieving the message \mathbf{m} and the average probability of error are approximately

$$\lambda_{\mathbf{m}} = p(\hat{\mathbf{m}} \neq \mathbf{m} \mid \mathbf{m}) \simeq p\left(w_H(\mathbf{e}^{\text{DBF}}) > \lfloor \frac{d_{\min}-1}{2} \rfloor \mid \mathbf{m}\right), \quad (5.19)$$

and

$$p_e^{(N)} \simeq \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p\left(w_H(\mathbf{e}^{\text{DBF}}) > \lfloor \frac{d_{\min}-1}{2} \rfloor \mid \mathbf{m}\right), \quad (5.20)$$

respectively. In the following remark, we discuss the channel noiseless assumption after removing harmful configurations.

Remark 4 *The theory of constrained coding began with Claude Shannon's classical 1948 paper [28], "A Mathematical Theory of Communications." In his setting, the channel "seen" by a constrained encoder/decoder is noiseless. Strictly speaking, this is not a realistic assumption because constrained coding is in practice used on noisy channels. In other words, even if the constraint is satisfied, bits can be in error. The probability of error is thus data-dependent. This assumption which is also used here is a generalization of the assumption made in Shannon's paper.*

Now, the goal is to minimize the average probability of error in 5.20. There may be different choices of deliberate errors \mathbf{e}^{DBF} that can remove the harmful

configurations from the input pattern, but some of them may exceed error correction capability of the code. The first challenge is to not overburden the decoder with inverting tiles more than the number of errors that the decoder can correct. Ideally, the tile selection function needs only to search for deliberate error patterns with Hamming weight lying within the error correction capability of the code being used. However, there may exist an input pattern/patterns where the number of deliberate bit errors required for removing harmful configurations exceeds the error correction capability of the code. Therefore, the coding method in this case might not be capacity achieving, and the probability of error correspondingly might be non-zero for some input patterns. The second challenge of using the deliberate error insertion method is to find the error pattern which has the minimum Hamming weight among the error patterns that can remove the harmful configurations, or, equivalently, $w_H(\mathbf{e}^{\text{DBF}})$ should be minimized for each message $\mathbf{m} \in \mathcal{M}$. Therefore, the roles of the tile-selection function θ are (i) to identify and remove the harmful configurations $\mathcal{X}_{\mathcal{P}_{i,j}}^B$ from a given input pattern and (ii) to find the error pattern which can remove the harmful configurations and has the minimum Hamming weight. It is worth mentioning that the overall performance of system is a function of d_{\min} of the code being used and depends on the choice of ECC, not the DBF method by itself. In the following, we characterize the role of tile-selection function θ .

For the input pattern \mathbf{x} , let $\mathcal{E}^{\mathbf{x}}$ be the set of all error patterns that can remove the \mathcal{P} -shaped configurations from the input pattern \mathbf{x} , i.e.,

$$\mathcal{E}^{\mathbf{x}} = \{ \mathbf{e}^{\text{DBF}} | \hat{\mathbf{x}} = \mathbf{x} \oplus \mathbf{e}^{\text{DBF}} \in \mathbb{S} \}. \quad (5.21)$$

In order to minimize the average probability of error in 5.20, we need to find

an error pattern $\mathbf{e}_{\text{DBF}}^*$ which has the minimum Hamming weight among the error patterns in \mathcal{E}^x , or another word,

$$\mathbf{e}_{\text{DBF}}^* = \arg \min_{\mathbf{e}_{\text{DBF}} \in \mathcal{E}^x} \{w_H(\mathbf{e}^{\text{DBF}})\}. \quad (5.22)$$

This problem can be regarded as a combinatorial optimization problem in which one needs to find an array \mathbf{e}^{DBF} minimizing $w_H(\mathbf{e}^{\text{DBF}})$ subject to the constraint that $\mathbf{e}^{\text{DBF}} \in \mathcal{E}^x$.

In the following, we provide examples of BCH-[15, 5, 7] codewords that are arranged into 3×5 arrays, as they help to explain the concepts we have introduced so far. We want to characterize the above constrained minimization problem for removing forbidden configurations by 2-D n.i.b. constraint from the 2-D arrays.

Example 2 *We assume that the user messages are the following binary vectors of length 5, $\mathbf{m}_1 = (0, 1, 0, 0, 0)$, $\mathbf{m}_2 = (1, 0, 0, 0, 0)$, $\mathbf{m}_3 = (0, 1, 1, 1, 1)$ and $\mathbf{m}_4 = (0, 1, 1, 0, 1)$, and are encoded by the triple-error correcting BCH-[15, 5, 7] code. We have the codewords*

$$\begin{aligned} \mathbf{c}_1 &= (0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0), \\ \mathbf{c}_2 &= (1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1), \\ \mathbf{c}_3 &= (0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0), \\ \mathbf{c}_4 &= (0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0), \end{aligned} \quad (5.23)$$

of length 15 which are then arranged into 3×5 arrays as four different patterns. The patterns are shown in Fig. 5.5, where the first row of each pattern is equipped with its corresponding user message. We only consider these four patterns out of 32 possible patterns by BCH-[15, 5, 7] code as they cover

all different tile colors inverting scenarios using the deliberate error insertion method.

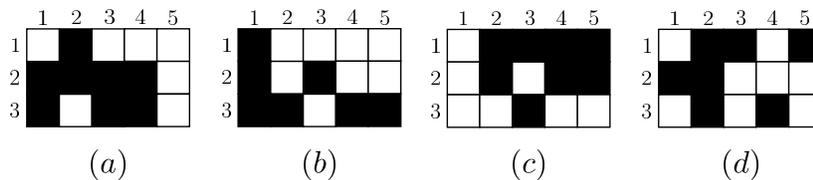


Figure 5.5: The input patterns for Example 2. We assume white tiles (zero entries) outside of each input pattern.

We are interested in removing 2-D isolated-bits configurations entirely from the above patterns with inverting colors of minimal number of tiles. In other words, the goal is to find the error pattern \mathbf{e}^{DBF} for each input pattern \mathbf{x} which has the minimum Hamming weight and $\mathbf{x} \oplus \mathbf{e}^{DBF}$ does not contain any of the 2-D isolated-bits configurations. Therefore, we have

$$\mathbf{e}_{(a)}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{e}_{(b)}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{e}_{(c)}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{e}_{(d)}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5.24)$$

In Fig. 5.5(a), the pattern does not contain any of the 2-D isolated-bits configurations, therefore there is no need to invert the tile colors, and $w_H(\mathbf{e}_{(a)}) = 0$. The pattern in Fig. 5.5(b) contains only one 2-D isolated-bits pattern, which is $\mathbf{x}_{Q^+(2,3)}$. One can remove this 2-D isolated-bits pattern by inverting the color of any one of the tiles in $Q^+(2,3)$, and therefore $w_H(\mathbf{e}_{(b)}) = 1$. For the pattern in Fig. 5.5(c), there are two overlapping 2-D isolated-bits pat-

terns, which are $\mathbf{x}_{Q^+(2,3)}$ and $\mathbf{x}_{Q^+(3,3)}$. These two isolated-bits patterns can be removed simultaneously by inverting either the color of tile (2, 3) or (3, 3), and therefore for this case also $w_H(\mathbf{e}_{(c)}) = 1$. In Fig. 5.5(d), the pattern contains two non-overlapping 2-D isolated-bits patterns, which are $\mathbf{x}_{Q^+(1,5)}$ and $\mathbf{x}_{Q^+(3,4)}$. One needs to invert at least colors of two tiles over this input pattern, and for this case $w_H(\mathbf{e}_{(d)}) = 2$. For the systematic BCH-[15, 5, 7] code (where the code-words are arranged into 3×5 arrays and the first row is equipped with the user bits), in average it needs to flip 0.6563 bits/pattern to remove the forbidden configurations by the 2-D n.i.b. constraint.

In the following, we provide remarks on the difficulty of the constrained minimization problem in the DBF method, and the difference of this method with conventional constrained coding methods.

Remark 5 *Finding the error pattern which removes a given set of 2-D configurations from a 2-D pattern and has the minimum Hamming weight via an exhaustive search among all admissible error patterns can be computationally prohibitive for large patterns. The above deliberate error insertion method can be regarded as a procedure for finding the minimum number of inversion operations required for converting a binary pattern to another binary pattern which does not contain any of channel forbidden configurations. This problem can be considered as a sub-class of Levenshtine distance problem [89], which is known as a hard combinatorial problem.*

Remark 6 *It is worth mentioning that problems related to 2-D constrained coding are in general difficult, as mainly it is hard to enumerate the patterns satisfying a 2-D constraint and having a uniform distribution, or, achieving the Shannon's noiseless channel capacity of the constraint. Let's denote this set of uniformly distributed patterns which satisfy the constraint by \mathcal{S} . The*

probability distribution achieving the 2-D noiseless channel capacity (or the maximum entropy of constraint) is

$$p(\hat{\mathbf{x}}) = \begin{cases} \frac{1}{|\mathbb{S}|}, & \hat{\mathbf{x}} \in \mathbb{S}, \\ 0, & \text{other.} \end{cases} \quad (5.25)$$

Therefore, the patterns in the set \mathbb{S} are equiprobable. In our method, instead of enumerating the patterns in \mathbb{S} (the way of conventional constrained coding methods), for a given input pattern \mathbf{x} (which may or may not be in \mathbb{S}), we try to find an $\hat{\mathbf{x}} \in \mathbb{S}$ which minimizes $w_H(\mathbf{x} \oplus \hat{\mathbf{x}})$.

In the following section, we reformulate this minimization problem with a probabilistic graphical formulation to cater the possibility of using message passing algorithms for finding approximate solutions.

5.4 A Probabilistic Graphical Formulation for Minimizing Bit Flips

In this section, we devise a probabilistic graphical formulation for the problem of minimizing the number of bit flips in the DBF method. The probabilistic graphical model of the problem defines a uniform distribution over \mathbb{S} where each pattern containing any of harmful configurations has zero probability. In this framework, the Hamming distance metric is translated with Binomial expression, and for a given input pattern \mathbf{x} , the constrained minimization problem becomes a 2-D maximum *a posteriori* problem. We use GBP, as a MAP inference method, to find approximate solution for marginal probabilities with minimizing the Bethe free energy (using the region based approximation method), and therefore an approximate solution for the problem of minimizing the number of flipped bits in the DBF scheme.

For a given binary pattern $\mathbf{x} \in \mathcal{X}$, the problem is to find an assignment,

$\hat{\mathbf{x}} \in \mathbb{S}$, that has the minimum Hamming distance with \mathbf{x} , or, equivalently, minimizes $w_H(\hat{\mathbf{x}} \oplus \mathbf{x})$. Since $w_H(\mathbf{x} \oplus \mathbf{x}) = 0$, if the pattern $\mathbf{x} \in \mathbb{S}$, the optimal answer is \mathbf{x} itself, i.e., there is no need to flip bits in \mathbf{x} . For the case $\mathbf{x} \notin \mathbb{S}$, we need to calculate the Hamming distance between each $\hat{\mathbf{x}} \in \mathbb{S}$ and \mathbf{x} , which can be intractable for large pattern. As it can be verified for each tile (i, j) locally over a finite neighborhood of tiles $\mathcal{P}_{i,j}$ whether the tile is contained in a harmful pattern of the set $\mathcal{X}_{\mathcal{P}_{i,j}}^B$, we define a *local distortion* function D for each tile (i, j) over $\mathcal{P}_{i,j}$ to compute the Hamming distance between different $\hat{\mathbf{x}} \in \mathbb{S}$ and the given input \mathbf{x} locally as follows. For every tile $(i, j) \in \mathcal{A}_{m,n}$, the function $D : \{0, 1\}^{|\mathcal{P}_{i,j}|} \times \{0, 1\}^{|\mathcal{P}_{i,j}|} \rightarrow \mathbb{N}$ is defined over the tiles indexed by $\mathcal{P}_{i,j}$ as follows

$$D(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}, \mathbf{x}_{\mathcal{P}_{i,j}}) = \begin{cases} w_H(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \oplus \mathbf{x}_{\mathcal{P}_{i,j}}), & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}_{i,j}}^B, \\ \infty, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B, \end{cases} \quad (5.26)$$

where $w_H(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \oplus \mathbf{x}_{\mathcal{P}_{i,j}})$ is the Hamming distance between $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}$ and $\mathbf{x}_{\mathcal{P}_{i,j}}$, and the patterns belonging to the set of harmful patterns are specified by ∞ . We should note that there can be different configurations of $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}_{i,j}}^B$ which have the same Hamming distance with $\mathbf{x}_{i,j}$. One may use the outputs of D for the tiles $(i, j) \in \mathcal{A}_{m,n}$ to find $\mathbf{x}^* \in \mathbb{S}$ which has the minimum Hamming distance with \mathbf{x} . This process can be intractable for large patterns as it needs to compute the output of D for every tile $(i, j) \in \mathcal{A}_{m,n}$, which has $2^{|\mathcal{P}_{i,j}|}$ different configurations, and take exponentially large memory just to store. In the following, we present a probabilistic formulation using a graphical model to find approximate solution for this problem using the GBP algorithm.

In order to present a probabilistic formulation for the distortion indicator function defined in (5.26), we use the binomial expression to translate

the Hamming distance metric into the probability domain. We assume that the color of each tile contained in a harmful configuration is inverted with the probability $0 < \lambda \leq 1$. For every tile $(i, j) \in \mathcal{A}_{m,n}$, we define a function $D_p : \{0, 1\}^{\mathcal{P}_{i,j}} \times \{0, 1\}^{\mathcal{P}_{i,j}} \rightarrow \mathbb{R}^{[0,1]}$ over the tiles indexed by $\mathcal{P}_{i,j}$,

$$D_p(\mathbf{x}_{\mathcal{P}_{i,j}}, \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}) = \begin{cases} \lambda^{w_H(\mathbf{e}_{\mathcal{P}_{i,j}})} (1 - \lambda)^{|\mathcal{P}_{i,j}| - w_H(\mathbf{e}_{\mathcal{P}_{i,j}})}, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}_{i,j}}^B, \\ 0, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B, \end{cases} \quad (5.27)$$

where $\mathbf{e}_{\mathcal{P}_{i,j}} = \hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \oplus \mathbf{x}_{\mathcal{P}_{i,j}}$ and $|\mathcal{P}_{i,j}|$ indicates the number of tiles in $\mathcal{P}_{i,j}$. This function is called as the *local probabilistic distortion* function. For each tile $(i, j) \in \mathcal{A}_{m,n}$, the distortion now is defined as the probability of having a distorted pattern $\mathbf{x}_{\mathcal{P}_{i,j}}$ which has the Hamming distance $w_H(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \oplus \mathbf{x}_{\mathcal{P}_{i,j}})$ with $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}_{i,j}}^B$. When $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B$, this probability is zero, as we are looking for patterns which do not belong to the set of harmful patterns. For a given input pattern \mathbf{x} and a set of forbidden patterns $\mathcal{X}_{\mathcal{P}_{i,j}}^B$, we are now interested in finding $\hat{\mathbf{x}} \in \mathbb{S}$ maximizing $p(\hat{\mathbf{x}}|\mathbf{x})$, which is equivalent to finding $\hat{\mathbf{x}}$ that minimizes $w_H(\hat{\mathbf{x}} \oplus \mathbf{x})$. In another word, we want to find

$$\hat{\mathbf{x}} = \arg \max_{\hat{\mathbf{x}} \in \mathbb{S}} \{p(\hat{\mathbf{x}}|\mathbf{x})\}. \quad (5.28)$$

The *a-posteriori* probability $p(\hat{\mathbf{x}}|\mathbf{x})$ for a fixed λ is

$$\begin{aligned} p(\hat{\mathbf{x}}|\mathbf{x}) &= \frac{p(\mathbf{x}|\hat{\mathbf{x}})p(\hat{\mathbf{x}})}{p(\mathbf{x})} \stackrel{(a)}{\propto} p(\mathbf{x}|\hat{\mathbf{x}}) \stackrel{(b)}{=} \prod_{(i,j) \in \mathcal{A}_{m,n}} p(\mathbf{x}_{i,j}|\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}), \\ &\stackrel{(c)}{=} \prod_{(i,j) \in \mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}} (1 - \lambda)^{1 - \mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}}, \end{aligned} \quad (5.29)$$

where (a) comes from this fact that the *a-priori* probability of choosing each pattern $\hat{\mathbf{x}} \in \mathbb{S}$ is equiprobable, (b) is established as for each tile (i, j) we can determine locally over $\mathcal{P}_{i,j}$ that the tile is contained in a harmful pattern, and (c) is obtained based on the definition of the local probabilistic distortion function, given in (5.27). Therefore, we have

$$p(\hat{\mathbf{x}}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{(i,j) \in \mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}} (1 - \lambda)^{1 - \mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}}, \quad (5.30)$$

where the normalization constant $Z(\mathbf{x})$, so called the partition function, is given by

$$Z(\mathbf{x}) = \sum_{\hat{\mathbf{x}} \in \{0,1\}^{m \times n}} \prod_{(i,j) \in \mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}} (1 - \lambda)^{1 - \mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}_{i,j}}^B\}}. \quad (5.31)$$

In order to compute the *a-posteriori* probability $p(\hat{\mathbf{x}}|\mathbf{x})$ with the factorization given in (5.30), we need to calculate the partition function given in the equation (5.31). Providing either exact or approximate solutions for the partition function in general is a NP-hard problem [7]. In [1] and [33], it is shown that the region-based approximation (RBA) method provides an approximate solution for the partition function by minimizing the region-based free energy (as an approximation to the variational free energy). In Appendix A, we first define a factor graph representation for the problem (maximizing $p(\hat{\mathbf{x}}|\mathbf{x})$ in (5.30) for a given input pattern \mathbf{x} subject to the constraint that $\hat{\mathbf{x}} \in \mathbb{S}$) and then formulate the RBA scheme for finding an approximate solution for this constrained maximization problem.

The following remarks discuss the optimality of the GBP-guided DBF method and the theoretical guarantee on the existence of solutions for the maximization problem given in (5.28).

Remark 7 For a given input pattern \mathbf{x} , we should note that the zero probability in (5.27) ensures that an approximate solution $\hat{\mathbf{x}}$ does not contain any harmful configurations, i.e., $\hat{\mathbf{x}} \in \mathbb{S}$. However, the approximate solution might not necessarily be the optimal pattern which minimizes $w_H(\hat{\mathbf{x}} \oplus \mathbf{x})$.

Remark 8 The problem of minimizing the number of bit flips in the DBF method can be considered as an instance of a constraint satisfaction problem (CSP). Statistical physicists consider different geometries of the solution space for a given CSP based on the density of constraint, which is defined as the ratio of the number of constraints to the number of variables. This density of constraint identifies satisfiability thresholds for the solution space of CSPs [11–15]. For the minimization problem in the DBF method for removing channel harmful configurations from an input pattern of a specific size, if the density of constraint lies in the satisfiable regions, then we can assume that there exist optimal solution/solutions for the problem.

5.5 Numerical Results

In this section, we present numerical analyses of the GBP-based DBF method for removing harmful patterns. Without loss of generality, we focus on the 2-D isolated-bits configurations in all our experiments. We first present the analysis on statistics of the number of flipped bits for removing 2-D isolated-bits patterns from random 2-D patterns. Furthermore, we study the convergence of the GBP algorithm as a function of the number of GBP iterations for different values of λ , the probability of flipping a bit in $\mathbf{x}_{\mathcal{P}_{i,j}}$ for $(i, j) \in \mathcal{A}_{m,n}$ which is defined in (5.27). To illustrate the usefulness of DBF method, we investigate its performance over the data-dependent channel in Section 5.2 under different scenarios in terms of the probability of uncorrectable bit errors, where the harmful configurations for the channel are the 2-D isolated-bits patterns. Fi-

nally, we compare the performance of the DBF method on a memoryless BSC with the row-by-row and bit-stuffing constrained coding schemes for the 2-D n.i.b. constraint, presented in [77] and [60] respectively.

Remark 9 *It should be noted that the parent-to-child message passing steps ([1]) in the GBP algorithm with considering all the regions for removing 2-D isolated-bits configurations operates with reasonable speed and memory requirements on binary patterns with maximum size of 32×32 . Thus in practice, the system would process these 32×32 (or smaller) tiles in a sequential way. As long as the scalability of method is concerned, the GBP algorithm can be implemented in a parallel fashion to work on multiple 32×32 binary patterns simultaneously.*

5.5.1 Statistics of The Number of Bit Flips for Removing 2-D Isolated-Bits Patterns

The performance of the DBF method relies on the error correction capability of the code being used, and of course the number of deliberate bit errors. Therefore, it is necessary to find how many bits in average are flipped within a codeword, and how this number compares to the error correction capability of the code. We have extracted the statistics of the number of bit flips for removing 2-D isolated-bits patterns from random 2-D patterns by the DBF method. In Fig. 5.6, we present an approximation of the occurrence probability of bit flipping, $p(w_H(\mathbf{e}^{\text{DBF}}))$, as a function of the number of flipped bits, $w_H(\mathbf{e}^{\text{DBF}})$. The statistics of number of flipped bits is obtained by using DBF for removing 2-D isolated-bits patterns from a sample set of 8000 random binary patterns of size 32×32 . Throughout all the simulations, we assume zero entries outside of random patterns. The average number of flipped bits is obtained by taking the average over all observed numbers of flipped bits,

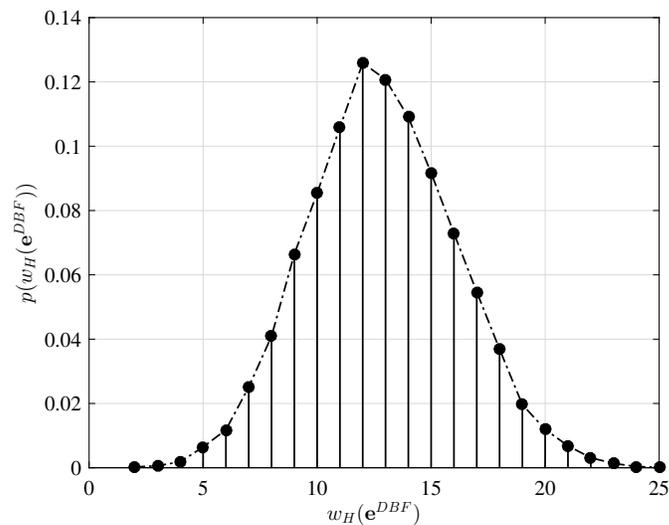


Figure 5.6: An approximation of the occurrence probability of bit flipping for removing the forbidden patterns by the 2-D n.i.b. constraint from random 32×32 arrays are given over 8000 trials. For this experiment, $\lambda = 0.1$ in (5.28).

which is $\overline{w_H(\mathbf{e})} = 12.84$. Therefore, approximately, it needs in average 12.84 bit flips in a random 32×32 pattern to remove the 2-D isolated-bits patterns. As long as the number of deliberate bit errors lies within the error correcting capability of an ECC, the codeword is guaranteed to be corrected. Using the occurrence probability of bit flipping, we can obtain the *uncorrectable bit error rate* (UBER) for an ECC used to correct these deliberate errors on a noiseless channel as follows

$$\text{UBER} = \left[\sum_{w_H(\mathbf{e}^{DBF}) > \lfloor \frac{d_{\min} - 1}{2} \rfloor} p(w_H(\mathbf{e}^{DBF})) \right] / NR, \quad (5.32)$$

where d_{\min} is the minimum distance of code, $N = m \times n$ is the size of the pattern (length of the code), and R is the rate of the ECC. Using BCH codes of length 1024 for correcting deliberate errors introduced in random 32×32 binary patterns for removing the 2-D isolated-bits configurations, the UBER is

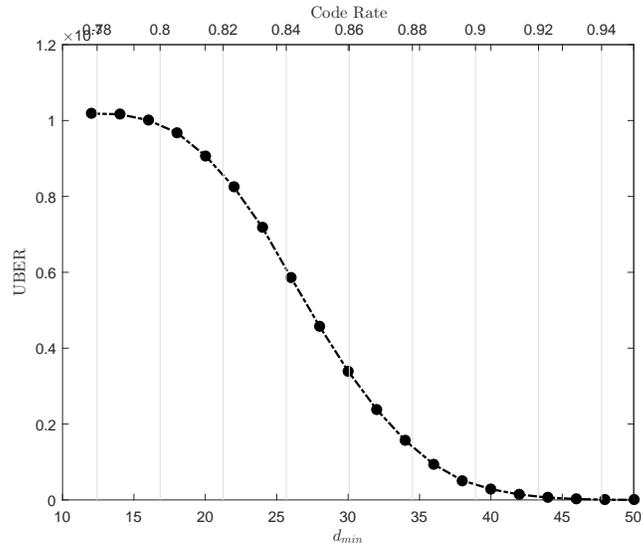


Figure 5.7: BCH codes of length 1024 with different code rates are used to correct the deliberate errors introduced in random 32×32 patterns for removing 2-D isolated-bits patterns. Using the flipping probabilities in Fig. 5.6 and (5.32), the UBER is calculated for BCH codes of length 1024 with different rates (and consequently d_{\min}).

given as a function of d_{\min} in Fig. 5.7. This figure shows UBER corresponding to different code rates (and consequently d_{\min}) supported by the BCH code of length 1024.

The choice of λ in the probabilistic formulation of problem, (5.28), depends on the constraint and the underlying method for solving the minimization problem. Note that λ is not a critical parameter in the DBF method. However, it should be chosen to be in the convergence region of GBP. As an example, we present the convergence of the GBP algorithm for finding the optimal error pattern to remove 2-D isolated-bits patterns from random 32×32 binary arrays for different values of λ . Fig. 5.8 shows the average number of flipped bits as a function of the number of iterations for different values of λ . It can be seen that convergence behaviors of the GBP algorithm for $\lambda \in \{0.04, 0.1, 0.18\}$ are very similar, and it is only the matter of choosing a λ that lies within the convergence region of the GBP algorithm. Throughout all our experiments in

this paper $\lambda = 0.1$, and the number of iterations for the GBP algorithm is 50 for 2-D isolated-bits patterns.

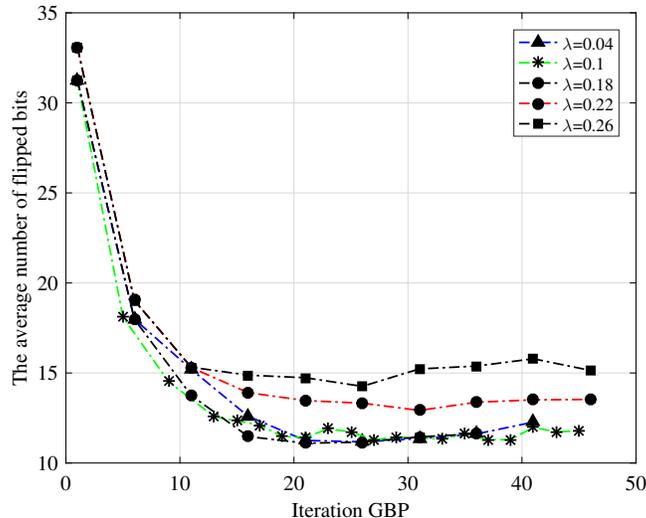


Figure 5.8: The average number of flipped bits for removing 2-D isolated-bits patterns from a random 32×32 array for different $\lambda \in \{0.04, 0.1, 0.18, 0.22, 0.26\}$ over 1000 trials versus the number of GBP iterations.

5.5.2 Performance Evaluation of The GBP-Guided DBF Method

In this section, we investigate the usefulness of DBF method for data-dependent 2-D channels, where specific patterns in channel inputs are the main cause of errors. We consider the introduced channel in Section 5.2 with the 2-D isolated-bits patterns as the harmful patterns for channel. For different values of α_b and α_g , we compare the average probability of error with and without incorporating the DBF method.

The user message \mathbf{m} of length K is encoded via an ECC with rate $R = \frac{K}{N}$, and the codeword $\mathbf{c}(\mathbf{m})$ of length $N = m \times n$ is arranged into a 2-D array $\mathbf{x}(\mathbf{m})$ of size $m \times n$. Prior to transmission over the channel, the 2-D isolated-bits patterns are removed from the input pattern by flipping minimum number of bits. The transmitted pattern over the channel is now $\mathbf{x}(\mathbf{m}) \oplus \mathbf{e}^{\text{DBF}}$, and the

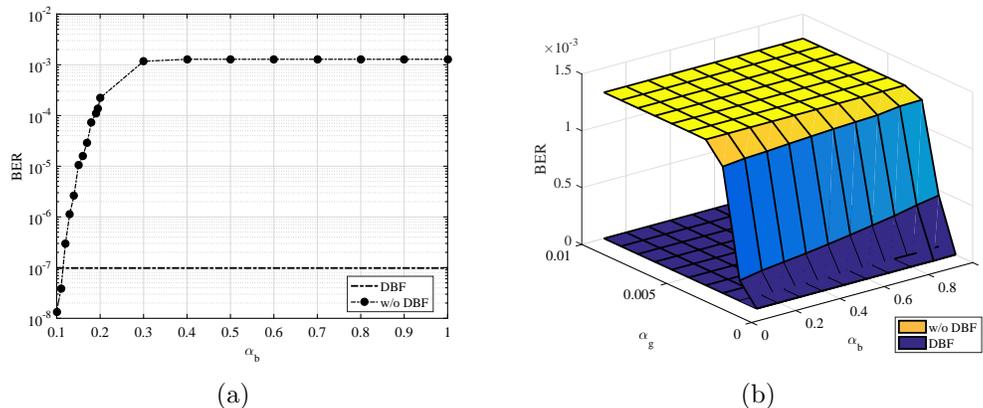


Figure 5.9: The average probability of error with and without incorporating for the cases (a) $\alpha_g = 0$ and $\alpha_b \in [0.1 : 0.1 : 1]$, and (b) $\alpha_g \in [0.001 : 0.001 : 0.01]$ and $\alpha_b = 100 \times \alpha_g$ is presented. In both cases the BCH-[1024, 728, 62] code is being used. The BER comparison results are obtained using the equations (33) and (34), and executing the GBP-guided DBF algorithm over at least 50,000 random instances of user messages.

received pattern is $\mathbf{x}(\mathbf{m}) \oplus \mathbf{e}^{\text{DBF}} \oplus \mathbf{e}^{\text{CH}}$. The transmitted pattern and channel output without DBF are $\mathbf{x}(\mathbf{m})$ and $\mathbf{x}(\mathbf{m}) \oplus \hat{\mathbf{e}}^{\text{CH}}$, respectively. Note that the channel is data-dependent, and therefore channel errors with and without incorporating DBF method are different. Using the bounded-distance decoder that can correct error patterns with Hamming weights lying within the error correction capability of the code, the average probability of error with and without incorporating the DBF method is simplified to

$$p_e^{(\text{DBF})} = \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p \left(w_H(\mathbf{e}^{\text{DBF}} \oplus \mathbf{e}^{\text{CH}}) > \lfloor \frac{d_{\min} - 1}{2} \rfloor | \mathbf{m} \right), \quad (5.33)$$

and

$$p_e^{(\text{w/o DBF})} = \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p \left(w_H(\hat{\mathbf{e}}^{\text{CH}}) > \lfloor \frac{d_{\min} - 1}{2} \rfloor | \mathbf{m} \right), \quad (5.34)$$

respectively, where d_{\min} is the minimum distance of the ECC.

In Fig. 5.9(a), we assume that channel errors solely come from appearances of 2-D isolated-bits configurations in input patterns, and $\alpha_g = 0$. Under this

assumption, removing the 2-D isolated-bits configurations from channel input patterns prior to transmission makes the channel noiseless. However without incorporating the DBF method, the color of tiles contained in a 2-D isolated-bits configuration invert with a probability of α_b . Therefore, the average probability of error with incorporating the DBF method for different values of α_g is constant. Fig. 5.9(a) shows the BER results with and without incorporating DBF for different values of α_b , when the BCH-[1024, 728, 62] code is used. It can be seen that for $0.3 \leq \alpha_b \leq 1$ we obtain approximately four orders of magnitude gain in the average BER with the GBP-guided DBF method. However, this gain is lower for smaller α_b 's as the number of deliberate bit errors introduced for removing 2-D isolated-bits configurations dominates the random channel bit errors. Fig. 5.9(b) shows the BER results with and without incorporating the GBP-guided DBF method, when $\alpha_g \in [0.001 : 0.001 : 0.01]$ and $\alpha_b = 100 \times \alpha_g$. This figure shows a reasonable gain in the BER performance with incorporating the GBP-guided DBF method.

5.5.3 Comparison Results on BSC

In this section, we compare the proposed scheme of imposing the 2-D n.i.b. constraint by deliberate errors against the row-by-row and the bit-stuffing coding schemes on a BSC. This can be interpreted as the case that 2-D isolated-bits configurations are the problematic patterns for the channel, and they must be removed before transmission, but removing these patterns does not make the channel noiseless. In our channel model, it is the case that $\alpha_b = 1$ and $\alpha_g \neq 0$. In the following, we first review the row-by-row and bit-stuffing methods for 2-D n.i.b. constraint and then present the comparison results.

Row-by-Row Coding Scheme for 2-D n.i.b. Constraint [77]

The encoder is a finite-state machine with 4 states, which maps each 3 information bits into a 2×2 binary pattern. For encoding information bits into an $m \times n$ array, strips of size $2 \times n$ are constructed using the encoded 2×2 binary patterns. Then, these strips are arranged in such a way to satisfy the 2-D n.i.b. constraint over the $m \times n$ array. The decoder is sliding-block decoder, where the decoding window size of the encoder is 3 bits.

Bit-Stuffing Scheme for 2-D n.i.b. Constraint [60]

The bit-stuffing method for mapping binary random sequences into a 2-D rectangular array satisfying the 2-D n.i.b. constraint is a variable rate coding scheme. First, the boundaries of the 2-D arrays are initialized with some fixed probability distribution. The encoding process has two steps. The encoder first generates two sequences with different statistics, Bernoulli(1/2) and Bernoulli(1/3), from the sequence of information bits using a probability transformer. Then, it encodes the unbiased and biased sequences into a 2-D array by inserting additional bits in such a way to ensure that the constraint is satisfied. At the decoder, the two sequences are recovered by doing the reverse process of inserting additional bits, and the binary sequence is recovered using an inverse probability transformer.

Raw BER Comparison Results

We compare the performance of the DBF method for imposing 2-D n.i.b. constraint into 2-D arrays of size 32×32 with the bit-stuffing and row-by-row constrained coding methods in terms of BER. It should be noted that the probability transformer in the bit-stuffing method is implemented in a one-to-one manner. Hence we can apply the reverse transformation to recover the

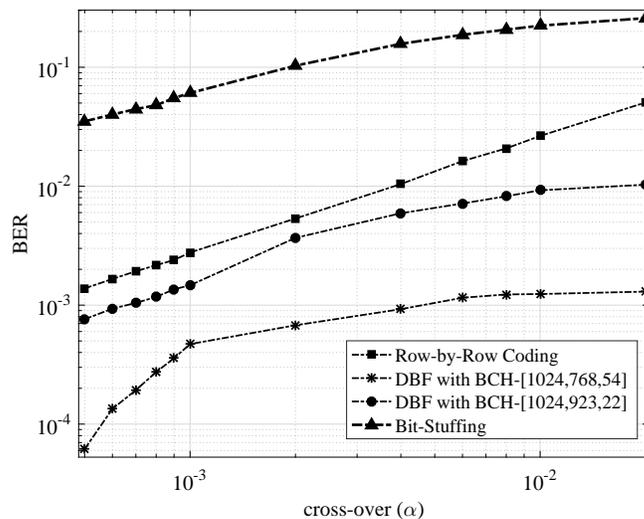


Figure 5.10: Figure shows the BER comparison results of the DBF, bit-stuffing and row-by-row coding methods on the BSC with the cross-over probability (α). The effect of error propagation can be observed in the BER curve of bit-stuffing which shows that this method is vulnerable to channel errors. The coding rate of DBF with BCH-[1024, 923, 22] code is close to the bit-stuffing method, and the rate of DBF with BCH-[1024, 768, 54] is close to the rate of row-by-row coding method.

original information bits. Fig. 5.10 shows the BER comparison results of the DBF, row-by-row and bit-stuffing methods over the BSC with the cross-over probability (α). It can be seen that the effect of error propagation in the row-by-row method is less severe than bit-stuffing as the row-by-row method uses a sliding-block decoder with error propagation window of 3 bits and the effective rate of 0.75. The average rate of bit-stuffing method for imposing 2-D n.i.b. constraint on a 32×32 array is $\simeq 0.91$. The bit-stuffing achieves a fairly high encoding rate for the 2-D n.i.b. constraint, but it suffers from the error propagation over noisy channels. The redundancy for imposing the constraint is now used in our scheme to strengthen the ECC (BCH code), resulting in a gain over the other schemes. For this purpose, we use the BCH-[1024, 923, 22] along with the DBF method for comparison with bit-stuffing method, and the DBF with BCH-[1024, 768, 54] for comparison with the row-by-row coding

method. We should note that we did not employ any forms of error correction in the row-by-row and bit-stuffing methods. Nevertheless, all the methods (including the DBF method with the BCH code) are designed to have the same overall coding rate.

CHAPTER 6

A Log-Likelihood Ratio based GBP for 2-D Channels

In order to improve throughput and energy consumption characteristics, as well as to obtain real time capabilities, hardware acceleration using dedicated architectures is employed for BP algorithms [90]. However, developing hardware architectures for GBP presents several challenges, due to the fact that the messages propagated among regions are conditional probabilities. These include: *(i)* divisions in message update equations, *(ii)* multiplication in both message and belief update equations, and *(iii)* requirements for very large precision, usually in floating point formats. In this paper, we propose a log-likelihood ratio (LLR) based GBP algorithm to address the hardware implementation issues by relying on only addition based operations (additions, subtractions and comparisons) with messages and beliefs represented in fixed point formats. This is achieved by introducing LLR based representations for messages and beliefs. The LLR representations allow us to devise arithmetic operations in log-likelihood domain for both message and belief update equations. The log-likelihood messages represent the standard approach in a wide range of iterative message-passing algorithms, including Turbo decoding [91], LDPC decoding - both binary [92] and non-binary [93], but far from trivial in inference algorithms such as GBP where messages express complex dependencies among variables. The proposed approach presents the following advantages: *(i)* divisions and multiplications are reduced in logarithm-domain to subtractions and additions; *(ii)* arithmetic operations are performed using fixed point formats, that has reduced complexity with respect to floating point

representations; *(iii)* the usage of ratios for decoding and detection problems lead to simple sign based hard decision mechanisms.

Several approaches to improve the computational parameters - processing time and memory requirements - of GBP have been proposed in [94–96]. These optimization techniques rely on two approaches: *(i)* reducing the number of arithmetic operations, by employing techniques such as result caching, conversion of a grid search into a linear search problem, or hierarchical state-space reduction [94, 95], and *(ii)* reducing the complexity of arithmetic operations for message and belief update equations, by performing them in logarithm-domain [96]. The latter targets elimination of divisions and multiplications, using only addition based operations. The proposed optimization target complexity reduction in the message and belief updates, targeted mainly for decoding and detection problems, performing the operations in logarithm-domain. With respect to [96], our main contributions are: *(i)* development of a ratio based version, and *(ii)* utilization of fixed point formats, instead of the more computationally complex floating point format.

We apply the proposed LLR-GBP for an image reconstruction application, denoising of images affected by a binary-input two-dimensional (2-D) Gaussian channel and additive white Gaussian noise (AWGN). Simulation results show that LLR-GBP with messages and beliefs represented in a 24-bit fixed point format, has similar performance to the floating point implementation. GBP as an image denoising algorithm works on probabilistic graphical model of the 2-D Gaussian channel with AWGN. There are many cycles in the factor graph representation of a 2-D Gaussian channel [37], which invalidates the tree-like assumption used in BP and leads to poor performance. In order to show that GBP can address the issues of short cycles in BP related methods, we also compare the performance of our LLR-GBP with JTED [35], that uses fixed point

formats, for detection of binary arrays passed through a 2-D intersymbol interference (ISI) channel. JTED can be considered as a sequential tree-reweighted sum-product algorithm [7], where for 2-D detection uses BCJR for computing exact marginals over row and column directions, and incorporates a message passing paradigm along both dimensions in an iterative manner for exchanging extrinsic information. However, this scheme still suffers from the cycles in the underlying graphical model of 2-D ISI channel for passing extrinsic information between row and column detectors. Our simulation results indicate that the reduced complexity LLR-GBP (with 24 bits, 8 bits fractional and 16 bits offset intervals) outperforms JETD with around 2 dB in terms of bit-error rate performance.

The chapter is organized as follows. We first present a detailed description of a constraint satisfiability problem (CSP). Section 6.2 is dedicated to the log likelihood GBP version. The experiment setup for image denoising over 2-D ISI Gaussian channel is explained in Section 6.3. Finally, simulation results and discussions are presented in 6.4.

Throughout this chapter, we denote the set of integers $\{n_1, n_1 + 1, \dots, n_2\}$ by $[n_1 : n_2]$ and the set of real numbers between n_1 and n_2 by (n_1, n_2) .

6.1 Constraint Satisfiability Problem

A CSP is defined by a set of N variables $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$ and a set of M constraints $\mathbf{C} = \{C_1, C_2, \dots, C_M\}$. Each variable X_i takes values x_i from a discrete and finite alphabet \mathcal{X} so that an assignment to the variables $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \mathcal{X}^N$. Let us assume that each constraint contains K variables. We denote the set of variables involving in the constraint C_i by \mathbf{X}_{C_i} and realizations of these variables by \mathbf{x}_{C_i} . The constraint C_i is characterized by the function $f_{C_i} : \mathcal{X}^K \rightarrow \{0, 1\}$ which specifies allowable combinations of

the values for the subset of variables participating in the constraint C_i such that the constraint C_i is satisfied if $f_{C_i}(\mathbf{x}_{C_i}) = 1$. A solution to a CSP is an assignment to all variables $\mathbf{x} = (x_1, x_2, \dots, x_N)$ that satisfies all M constraints.

The set of assignments to variables satisfying a CSP is identified by

$$\mathcal{S}_{\mathbf{C}} = \left\{ \mathbf{x} \in \mathcal{X}^N : \prod_{C_i} f_{C_i}(\mathbf{x}_{C_i}) = 1 \right\}. \quad (6.1)$$

We define a probability measure over this set of SAT assignments as follows

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C_i \in \mathbf{C}} f_{C_i}(\mathbf{x}_{C_i}), \quad (6.2)$$

where the normalization constraint Z , so called the partition function, is given by

$$Z = \sum_{\mathbf{x} \in \mathcal{X}^N} \prod_{C_i \in \mathbf{C}} f_{C_i}(\mathbf{x}_{C_i}). \quad (6.3)$$

In fact, $p(\mathbf{x})$ is the uniform probability distribution over the set $\mathcal{S}_{\mathbf{C}}$. The uniform distribution given in Eq. (6.2) is expressed in a sum-product form. Such factorization is known to satisfy certain properties called Markovian properties and the corresponding graphical model is a Markov random field.

Many of inference problems in computer vision, error-correction coding and artificial intelligence can be reformulated as the computation of marginal probabilities of a joint probability distribution over the set of SAT assignments [5, 6, 97]. This is equivalent to finding the fraction of satisfying assignments in which a variable is assigned a particular value. Given a joint distribution $p(\mathbf{x}) = p(x_1, x_2, \dots, x_N)$, the marginal distribution of a subset of variables \mathbf{x}_S , where $S \subset [1 : N]$, is the probability distribution of variables \mathbf{x}_S averaging over all information about $\mathbf{x} \setminus \mathbf{x}_S$. This can be calculated by

summing $p(x_1, x_2, \dots, x_N)$ over $\mathbf{x} \setminus \mathbf{x}_S$, i.e.,

$$p(\mathbf{x}_S) = \sum_{\mathbf{x} \setminus \mathbf{x}_S} p(x_1, x_2, \dots, x_N). \quad (6.4)$$

This process of computing marginal probability distributions can be intractable for large n as it needs to take summation over exponential number of values of variables.

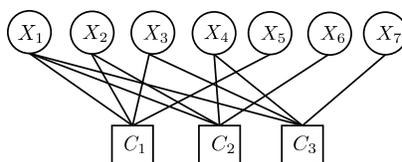


Figure 6.1: The factor graph for the joint probability distribution in the Eq. (6.5) is given. The set of variable nodes $\mathbf{X} = \{X_1, X_2, \dots, X_7\}$ represents the error patterns and the set of factor nodes $\mathbf{C} = \{C_1, C_2, C_3\}$ verify the syndrome constraints.

Graphical models provide an intuitive framework for representing interacting sets of variables and constraints. Using the factor graph formalism [6], a CSP can be described by a bipartite graph $G = (\mathbf{X} \cup \mathbf{C}, \mathbf{E})$ with two types of nodes, namely variable nodes \mathbf{V} and factor nodes \mathbf{F} , and a set of edges \mathbf{E} . Variables $X_i \in \mathbf{X}$ are symbolized by variable nodes; constraints $C_j \in \mathbf{C}$ are symbolized by factor nodes; and the dependence of a constraint on a variable is symbolized by an edge joining the two. We denote the variable nodes by circle nodes and the constraints by square nodes, where the edge (X_i, C_j) between the factor node C_j and the variable node X_i is included in \mathbf{E} if and only if $X_i \in \mathbf{X}_{C_j}$. The set of variable nodes connected to the factor node C_j is denoted by \mathcal{N}_{C_j} and similarly the set of factor nodes connected to the variable node X_i is denoted by \mathcal{N}_{X_i} . As an example, a factor graph corresponding to

the following joint distribution

$$p(x_1, x_2, x_3, \dots, x_7) = \frac{1}{Z} f_A(x_1, x_2, x_3, x_5) f_B(x_1, x_2, x_4, x_6) f_C(x_1, x_3, x_4, x_7), \quad (6.5)$$

is given in Figure 6.1, where Z is some normalization constraint. Traditional low-complexity approximate algorithms for solving these problems are based on BP [8, 9] which operate on factor graphs. BP, as an algorithm to compute marginals of functions on a factor graph, has its roots in the broad class of Bayesian inference problems [10]. It is well known that the BP algorithm gives exact inference only on cycle-free graphs (trees). It has been also observed that in some applications the BP can provide close approximations to exact marginals on loopy graphs. However, an understanding of the behavior of BP in the latter case is far from complete. Moreover, it is known that BP does not perform well on graphs which contain a large number of short cycles. In the following section, we introduce a LLR-based GBP algorithm as a reduced complexity method for solving problems involving probabilistic inference.

6.2 Log-Likelihood Ratio based GBP Algorithm

Similar to the log-likelihood versions of BP [91, 92], as a first step to reduce the complexity of GBP, we define ratios for messages and beliefs. The ratio of beliefs for the region $R \in \mathcal{R}$ at iteration k is defined by

$$\beta_R^{(k)}(\mathbf{x}_R) = \frac{b_R^{(k)}(\mathbf{x}_R)}{b_R^{(k)}(\mathbf{x}_R^{\text{ref}})}, \quad (6.6)$$

where $\mathbf{x}_R^{\text{ref}}$ represents the reference state for the ratio-domain, and $b_R^{(k)}(\mathbf{x}_R^{\text{ref}})$ is the belief corresponding to this event. Similarly, the ratio of messages coming

to the region R from its parent regions $P \in \mathcal{P}_R$ at iteration k is determined by

$$\lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R) = \frac{m_{P \rightarrow R}^{(k)}(\mathbf{x}_R)}{m_{P \rightarrow R}^{(k)}(x_R^{\text{ref}})}, \quad (6.7)$$

where $m_{P \rightarrow R}^{(k)}(x_R^{\text{ref}})$ is the probability that the parent region $P \in \mathcal{P}_R$, at iteration k , sends a message to the region R that the state of its variables is the reference state. We have considered the all-one state (the state that all variables have value 1) as the reference state in our implementation.

Using the ratio of messages, the message update equation at iteration k becomes

$$\lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R) = \frac{\sum_{\mathbf{x}_{P \setminus R}} \prod_{F_{C_j} \in F_{P \setminus R}} \phi_{C_j}(\mathbf{x}_{C_j}) \prod_{(I,J) \in N(P,R)} \lambda_{I \rightarrow J}^{(k-1)}(\mathbf{x}_J)}{\left(\prod_{(I,J) \in D(P,R)} \lambda_{I \rightarrow J}^{(k-1)}(\mathbf{x}_J) \right) c_{P \rightarrow R}^{(k)}}, \quad (6.8)$$

where $\phi_{C_j}(\mathbf{x}_{C_j})$ is the ratio of constraint and $c_{P \rightarrow R}^{(k)}$ is the correction factor which ensures $\lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R^{\text{ref}}) = 1$. The ratio of constraint is defined by

$$\phi_{C_j}(\mathbf{x}_{C_j}) = \frac{f_{C_j}(\mathbf{x}_{C_j})}{f_{C_j}(\mathbf{x}_{C_j}^{\text{ref}})}, \quad (6.9)$$

where $f_{C_j}(\mathbf{x}_{C_j}^{\text{ref}})$ is value of function at the constraint C_j when the state of their variables, \mathbf{x}_{C_j} , is the reference state. The correction factor for messages from a parent region P to the region R is given by

$$c_{P \rightarrow R}^{(k)} = \sum_{\mathbf{x}_{P \setminus R}} \prod_{F_{C_j} \in F_{P \setminus R}} \phi_{C_j}(\mathbf{x}_{C_j}^{\text{ref}}) \prod_{(I,J) \in N(P,R)} \lambda_{I \rightarrow J}^{(k-1)}(\mathbf{x}_J^{\text{ref}}). \quad (6.10)$$

Furthermore, we have

$$\begin{aligned}\lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R) &= \lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R) \times \frac{1}{1 + \frac{1-\omega^{(k)}}{\omega^{(k)}} \times \frac{\sigma^{(k)}}{\sigma^{(k-1)}}} + \\ &\lambda_{P \rightarrow R}^{(k-1)}(\mathbf{x}_R) \times \frac{1}{1 + \frac{\omega^{(k)}}{1-\omega^{(k)}} \times \frac{\sigma^{(k-1)}}{\sigma^{(k)}}},\end{aligned}\quad (6.11)$$

where $\sigma^{(k)} = \sum_{\mathbf{x}_R} \lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R)$. The update of $\sigma^{(k)}$ is performed as follows

$$\sigma^{(k)} = \omega^{(k)} \sigma^{(k)} + (1 - \omega^{(k)}) \sigma^{(k-1)}.\quad (6.12)$$

Applying the logarithm, the multiplications in both belief and message update equations are reduced to additions, while the division in the message update equation becomes a subtraction. The message update equation (Eq. (6.8)) turns into

$$\begin{aligned}\Lambda_{P \rightarrow R}^{(k)}(\mathbf{x}_R) &= \underset{\mathbf{x}_{P \setminus R}}{\diamond} \left(\sum_{F_{C_j} \in F_{P \setminus R}} \Phi_{C_j}(\mathbf{x}_{C_j}) \sum_{(I,J) \in N(P,R)} \Lambda_{I \rightarrow J}^{(k-1)}(\mathbf{x}_J) \right) \\ &- \sum_{(I,J) \in D(P,R)} \Lambda_{I \rightarrow J}^{(k-1)}(\mathbf{x}_J) - C_{P \rightarrow R}^{(k)},\end{aligned}\quad (6.13)$$

where $\Lambda_{P \rightarrow R}^{(k)}$, Φ_{C_j} and $C_{P \rightarrow R}^{(k)}$, respectively, defined as the logarithm of $\lambda_{P \rightarrow R}^{(k)}$, ϕ_{C_j} and $c_{P \rightarrow R}^{(k)}$, $\diamond(\cdot)$ indicates the approximation used for computing the logarithm of the sum, ($\log(\sum)$) which is explained in the following.

Considering two positive real numbers $\lambda_1, \lambda_2 \in \mathbb{R}$, we have

$$\begin{aligned}\diamond(\lambda_1, \lambda_2) &= \log(\lambda_1 + \lambda_2) = \log(\max(\lambda_1, \lambda_2) + \min(\lambda_1, \lambda_2)), \\ &= \log(\max(\lambda_1, \lambda_2)) + \log\left(1 + \frac{\min(\lambda_1, \lambda_2)}{\max(\lambda_1, \lambda_2)}\right).\end{aligned}$$

We denote the term $\frac{\min(\lambda_1, \lambda_2)}{\max(\lambda_1, \lambda_2)}$ by η . According to the above equation, com-

putation of $\log(\sum)$ is reduced to a maximum and computation of $\log(1 + \eta)$. As $\lambda_1, \lambda_2 > 0$, $0 < \eta \leq 1$, and therefore $0 < \log(1 + \eta) \leq \log(2)$, we use the following method for approximating the term $\log(1 + \eta)$. We first split the $(0, 1)$ interval into k equal intervals as follows $(0, l_1), [l_1, l_2), \dots, [l_{k-1}, 1)$, where $l_i = \frac{1}{i \times k}$ and $i \leq k$. η is approximated with l_i , if $l_i \leq \eta < l_{i+1}$. In this method, we only need to perform k comparisons among η and l_i 's. In the logarithm-domain, the terms $\log(l_i)$ and $\log(1 + l_i)$ are constant and can be computed offline for a fixed number of intervals, k . A larger k allows better approximation at the expense of higher complexity.

6.3 Image Denoising over 2-D Gaussian Channels

In order to compare the performance of the proposed LLR based approach for GBP with the probability-domain floating point version, we use GBP for an image denoising application for reconstruction of images affected by 2-D Gaussian channels and independent noise, such as AWGN.

We assume that in all our experiments the size of Gaussian kernel is 3×3 . Let us denote the binary representation of an input image by an array $\mathbf{x} = [x_{i,j}]$, the kernel of Gaussian filters by \mathbf{H} , and the distorted version of input image by an array $\mathbf{y} = [y_{i,j}]$. We are interested in finding the most likely input samples $\hat{x}_{i,j}$ from \mathbf{y} . The (i, j) -th output sample, $y_{i,j}$, is the binary input affected by the 2-D Gaussian channel and is given by

$$y_{i,j} = \mathbf{H}\mathbf{x} [i, j] + n [i, j], \quad (6.14)$$

where

$$\mathbf{x}[i, j] = \begin{bmatrix} x_{i-1, j-1} & x_{i-1, j} & x_{i-1, j+1} \\ x_{i, j-1} & x_{i, j} & x_{i, j+1} \\ x_{i+1, j-1} & x_{i+1, j} & x_{i+1, j+1} \end{bmatrix} \quad (6.15)$$

and \mathbf{H} is represented the considered 3×3 Gaussian kernel, and $n[i, j]$ is a sample from a zero-mean and σ^2 -variance Gaussian distribution. The variance σ^2 is defined as a function of signal-to-noise ratio (SNR) so that

$$\sigma = \|\mathbf{H}\| \times 10^{-\text{SNR}/20}, \quad (6.16)$$

where SNR is given in db and $\|\cdot\|$ denotes the l_2 -norm.

The problem is to find the most likely input bits $\{x_{i,j}\}$ from \mathbf{y} that maximizes $p(x_{i,j}|\mathbf{y})$, for a fixed SNR value. The problem of maximizing these probabilities is reduced to computing

$$p(x_{i,j}|\mathbf{y}) \propto \sum_{\mathbf{x} \setminus x_{i,j}} \prod_{i,j} \exp\left(-\frac{(y_{i,j} - \mathbf{H}\mathbf{x}[i, j])^2}{2\sigma^2}\right). \quad (6.17)$$

The probabilities $\{p(x_{i,j}|\mathbf{y})\}$ are called *a posteriori* probabilities (APPs). Computing APPs is a hard problem as it requires to taking sum over exponential number of variables. We use the logarithmic likelihood ratio version of GBP for estimating APPs. The performane loss shows that the algorithm suffers from dependencies of messages and existense of cycles in the underlying graphical model for exchanging extrinsic information between row and column BCJR detector.

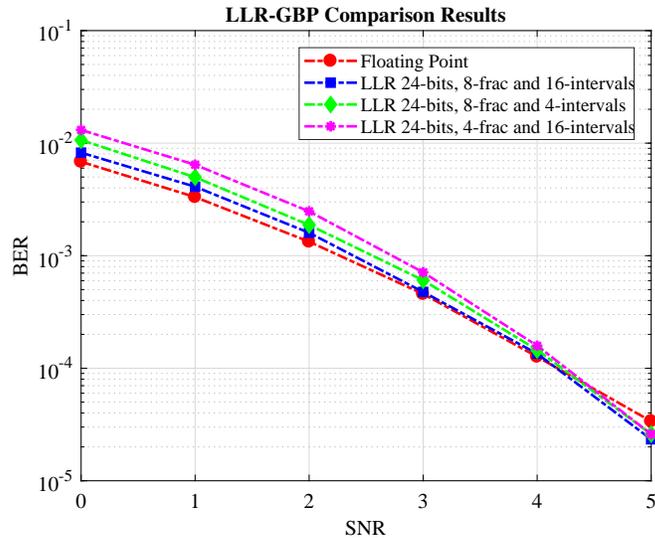


Figure 6.2: Detection performance curves of GBP for 64-bit double precision format, 24-bit fixed point LLR.

6.4 Simulation Results

We have applied GBP in both probability-domain, with messages and beliefs represented using 64-bits IEEE754 double precision floating point format, and in logarithm-domain using 24-bit fixed point format, with 4 and 8 bits for fractional part and with 4 and 16 offset constants in the approximation of $\log(\sum)$, for a SNR range of the AWGN noise from 0 to 5 db. The considered Gaussian kernel corresponds to a zero mean and a Results are plotted in Fig. 6.2. Fig. 6.2 indicates that the proposed LLR version has similar performance with respect to the floating point implementation, with a slight decrease in performance for low SNR regions (0-3 dB), and a slight increase in performance for higher SNR (5 dB). Reducing the number of bits associated with the fractional part will lead to a performance decrease. Furthermore, reducing the number of offset intervals in $\log(\sum)$ approximation will also impact the performance of the GBP. It is worth noted that reducing the number of bits associated to the fractional part does not lead to reduced computational complexity, while

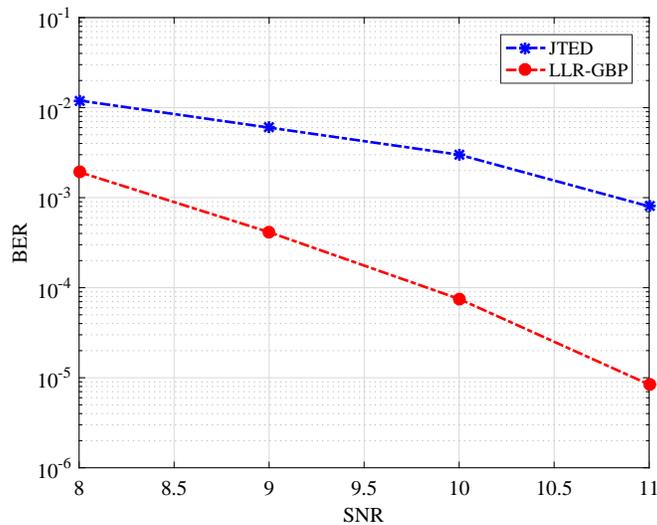


Figure 6.3: Comparison results between the proposed LLR-GBP (24-bit: 8 bits fractional and 16 bits offset intervals) and JTED.

reducing the number of offset intervals in the $\log(\sum)$ approximation will lead to reduced number of performed arithmetic operations (reduced number of comparisons with constants).

6.5 Comparison Results with JTED

In this subsection, we present the comparison results between the 24-bit fixed point LLR-GBP, with 8 bits for fractional and 16 offset intervals, and JTED proposed in [35] for detection of 2-D binary arrays passed through a 2-D ISI channel. The JTED method uses BCJR detectors [50], which give exact APPs for 1-D case, in row and column directions allowing the message passing along both dimensions in an iterative manner. The considered ISI channel has been

defined by

$$\mathbf{H} = \begin{bmatrix} 0.0625 & 0.25 & 0.0625 \\ 0.25 & 1 & 0.25 \\ 0.0625 & 0.25 & 0.0625 \end{bmatrix}. \quad (6.18)$$

We should note that, due to the computational complexity of the considered formulation of the GBP algorithm for detection, the maximum size of an input binary array can be 32×32 . For this, we have performed simulations on random 2-D binary arrays of size 32×32 for LLR-GBP, with respect to 64×64 random binary arrays for JTED [35]. Simulation results, presented in Fig. 6.3, indicate that the proposed LLR-GBP provides an almost 2 dB improvement in bit-error rate performance comparing with JTED.

CHAPTER 7

Conclusions

Recent advances in emerging data storage technologies like magnetic recording systems, optical recording devices and flash memory drives necessitate to study 2-D coding techniques for reliable storage of information. In these systems, user information bits are arranged into 2-D arrays for storing over the recording channel, and occurrences of specific patterns in input arrays are the significant cause of errors during read-back process. These systems require the use of some form of error-correction coding in addition to constrained coding of the input data or symbol sequences. It is therefore natural to investigate the interplay between these two forms of coding and the possibilities for efficiently combining their functions into a single coding operation. In this dissertation, we have focused on the problem of transmission of binary messages over data-dependent 2-D channels. Specifically, as on the prominent examples of data-dependent 2-D channels, we consider Two-Dimensional Magnetic Recording (TDMR) channels which is an emerging storage technology and achieves beyond 4 Tb/in². In TDMR, bit size and bit spacing are extremely small which leads to severe 2-D inter-symbol interference (ISI). TDMR uses only a small number of grains to store a bit of information. This reduction in the number of magnetic grains per bit leads to variations of bit boundaries, and consequently data dependent jitter noise. Neighboring bit transitions lead to an increased media noise which results in degradation of the detector performance. We have considered the following challenges in regard to the problem of reliable storage of binary messages over TDMR systems.

In Chapter 2, we have introduced a method to handle the media noise seen in a TDMR channel using a GBP based detector. We have used the GBP algorithm for signal detection in conjunction with the BP algorithm for LDPC decoding. In Chapter 3, we have identified the most harmful patterns in Voronoi based TDMR channels. In that work, we have concluded that the use of constrained codes can reduce the complexity of 2-D ISI signal detection since lesser 2-D ISI span can be accommodated at the cost of a nominal code rate loss. However, a system must be designed carefully so that the rate loss incurred by a 2D constraint does not offset the detector performance gain due to more distinguishable read-back signals. In Chapter 5, we have presented a deliberate bit flipping coding scheme for data-dependent 2-D channels. For this method, we have shown that the main obstacle is the number of deliberate errors which are introduced for removing harmful configurations before transmission through the channel. We have devised a combinatorial optimization formulation for minimizing the number of bit flips, and have explained how this problem can be related to a binary constraint satisfaction problem. Finally, through an example, we have presented uncorrectable bit-error rate results of incorporating DBF for removing 2-D isolated-bit configurations from 2-D patterns of certain size. We have evaluated the performance gain of our proposed approach on a data-dependent 2-D channel, where 2-D isolated-bits patterns are the harmful patterns for the channel. Furthermore, the performance of the DBF method is compared with classical 2-D constrained coding schemes for the 2-D *no isolated-bits* constraint on a memoryless binary symmetric channel.

In Chapter 6, we have proposed a log-likelihood ratio based GBP algorithm in order to reduce both the computational complexity and the storage requirements for GBP. We have demonstrated the validity of LLR-GBP on reconstruction of images passed through binary-input two-dimensional Gaussian

channels with memory and affected by additive white Gaussian noise. Simulation results performed for an image reconstruction application indicate that for 24-bit fixed point formats, a slight degradation in performance in low SNR regions (SNR 0 to 3) is obtained with respect to the 64-bit floating point probabilistic GBP. However, this slight degradation will come with improved storage requirements for the LLR version, with more than 2.5x reduction in storage for LLR based version. Reducing the number of fractional bits, as well as the number of offset constants used in the approximation of $\log(\sum)$, will reduce the detection performance in the low SNR regions.

Future Work

As a future work, the DBF method can be reformulated for 2-D semiconstrained coding. In some applications, we rather prefer not to remove entirely the harmful configurations, and we only want to limit the number of occurrences of specific configurations in a 2-D pattern. As in the case when the number of bit flips for imposing strong constraints is large and may overwhelm the ECC decoder, there is a need to allow some of the harmful configurations patterns to appear, yet not very often. For this purpose, the function D_p in (5.27) can be reformulated as a probability transformer function, which maps random binary patterns to binary patterns satisfying a desired empirical distribution for appearances of harmful configurations. The GBP algorithm still can be used to minimize the number of flipped bits for this mapping.

Quantum low-density parity check (QLDPC) codes are promising in realization of scalable, fault tolerant quantum memory for computation. Many of the QLDPC codes constructions suffer from unavoidable short cycles in their Tanner graph which degrade the decoding performance of the BP algorithm. As a future work, a syndrome based GBP algorithm for decoding of quantum

LDPC codes can be devised to escape from short cycle trapping sets compared to the BP algorithm. As another future work, GBP algorithm can be reformulated to find the most likely error coset to make use of degeneracy of quantum codes. Also, it would be interesting to find new trapping sets that adversely affect beliefs computed by GBP algorithm. Analyzing the complexity and also finding suitable trade-offs are also considered as our future work.

REFERENCES

- [1] J. Yedidia, W. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282 – 2312, Jul. 2005.
- [2] M. Ibnkahla, “Applications of neural networks to digital communications – a survey,” *Signal Processing*, vol. 80, no. 7, pp. 1185–1215, 2000.
- [3] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [4] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, “On deep learning-based channel decoding,” in *Proc. IEEE CISS*, Mar. 2017, pp. 1–6.
- [5] M. Jordan, *Learning in Graphical Models*. MIT Press, 1999.
- [6] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.
- [7] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends in Machine Learning*, vol. 1, pp. 1–305, Nov. 2008.
- [8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Francisco, CA: Kaufmann, 1988.
- [9] R. G. Gallager, “Low density parity check codes,” Ph.D. dissertation, Cambridge, MA, 1963.
- [10] B. J. Frey, *Graphical models for machine learning and digital communication*. Cambridge, MA, USA: MIT Press, 1998.
- [11] V. Chvatal and B. Reed, “Mick gets some the odds are on his side,” *Proc. of 33rd FOCS*, 1992.
- [12] O. Dubois, Y. Boufkhad, and J. Mandler, “Typical random 3-SAT formulae and the satisfiability threshold,” *Proc. of 11th SODA*, pp. 126–127, 2000.
- [13] E. Friedgut, “Necessary and sufficient conditions for sharp thresholds of graph properties and the k-problem,” *Journal of American Math Society*, vol. 12, pp. 1017–1054, 1999.

- [14] A. Goerdt, “A remark on random 2-SAT,” *Journal Computer System and Sciences*, vol. 53, pp. 469–486, 1996.
- [15] A. Kaporis, L. M. Kirousis, and E. G. Lalas, “The probabilistic analysis of a greedy satisfiability algorithm,” *Proc. of 10th Annual European Symp. on Algorithm*, pp. 574–585, 2000.
- [16] E. N. Maneva, E. Mossel, and M. J. Wainwright, “A new look at survey propagation and its generalizations,” *CoRR*, vol. cs.CC/0409012, 2004. [Online]. Available: <http://arxiv.org/abs/cs.CC/0409012>
- [17] S. Cook, “The complexity of theorem-proving procedures,” 1971, p. 151.
- [18] G. Parisi, “On local equilibrium equations for clustering states,” *CoRR*, vol. cs.CC/0212047, 2002. [Online]. Available: <http://arxiv.org/abs/cs.CC/0212047>
- [19] T. Mora, M. Mezard, and R. Zecchina, “Clustering of solutions in the random satisfiability problem,” *Phys. Rev. Lett.*, 2005.
- [20] M. Mezard, G. Parisi, and R. Zecchina, “Analytic and algorithmic solution of random satisfiability problems,” *Science*, 2002.
- [21] R. Kikuchi, “A Theory of Cooperative Phenomena,” *Physical Review Online Archive (Prola)*, vol. 81, no. 6, p. 988, Mar. 1951.
- [22] T. Morita, *Foundations and applications of cluster variation method and path probability method*. Publication Office, Progress of Theoretical Physics, 1994.
- [23] M. Khatami, V. Ravanmehr, and B. Vasić, “GBP-based detection and symmetric information rate for rectangular-grain TDMMR model,” in *in Proc. IEEE Int. Symp. Inf. Theory (ISIT 2014)*, Honolulu, Hawaii, June 29- -July 4 2014, pp. 701–705.
- [24] G. Sabato and M. Molkarai, “Generalized belief propagation for the noiseless capacity and information rates of run-length limited constraints,” *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 669–675, Mar. 2012.
- [25] J. Sibel, S. Reynal, and D. Declercq, “An application of generalized belief propagation: splitting trapping sets in LDPC codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, June 2014, pp. 706–710.
- [26] N. Raveendran, M. Bahrami, and B. Vasić, “Syndrome generalized belief propagation decoding for quantum memories (submitted),” in *Proc. IEEE Int. Conf. on Commun.*, 2019.

- [27] S. Shamai, L. H. Ozarow, and A. D. Wyner, “Information rates for a discrete-time gaussian channel with intersymbol interference and stationary inputs,” *IEEE Trans. on Inf. Theory*, vol. 37, no. 6, pp. 1527–1539, Nov 1991.
- [28] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [29] M. Khatami and B. Vasić, “Constrained coding and detection for tdmr using generalized belief propagation,” in *IEEE Int. Conf Commun. (ICC 2014)*, June 2014, pp. 3889–3895.
- [30] C. Matcha and S. G. Srinivasa, “Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR,” *IEEE Trans. Magn.*, vol. 51, no. 10, pp. 1–15, Oct 2015.
- [31] A. Krishnan, R. Radhakrishnan, B. Vasić, A. Kavcic, W. Ryan, and F. Erden, “2-D magnetic recording: Read channel modeling and detection,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3830–3836, Oct. 2009.
- [32] D. Dunbar and G. Humphreys, “A spatial data structure for fast poisson-disk sample generation,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 503–508, Jul. 2006. [Online]. Available: <http://doi.acm.org/10.1145/1141911.1141915>
- [33] P. Pakzad and V. Anantharam, “Estimation and marginalization using Kikuchi approximation methods,” *Neural Computation*, vol. 17, pp. 1836–1873, 2003.
- [34] M. Welling, “On the choice of regions for generalized belief propagation,” in *UAI '04*. Arlington, Virginia, United States: AUAI Press, 2004, pp. 585–592.
- [35] Y. Chen and S. G. Srinivasa, “Joint self-iterating equalization and detection for two-dimensional intersymbol-interference channels,” *IEEE Trans. Comm*, vol. 61, no. 8, pp. 3219–3230, Aug. 2013.
- [36] M. Khatami, M. Bahrami, and B. Vasić, “Symmetric information rate estimation and bit aspect ratio optimization for TDMR using generalized belief propagation,” in *IEEE Int. Symp. Inf. Theory*, Jun. 2015.
- [37] O. Shental, N. Shental, S. Shamai (Shitz), I. Kanter, A. Weiss, and Y. Weiss, “Discrete-input two-dimensional gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics,” *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1500–1513, Apr. 2008.

- [38] J. Chen and P. Siegel, "On the symmetric information rate of two-dimensional finite-state ISI channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 227–236, Jan 2006.
- [39] D. V. Nguyen, S. K. Chilappagari, B. Vasić, and M. W. Marcellin, "On the construction of structured LDPC codes free of small trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2280–2302, Apr. 2012.
- [40] R. E. Rottmayer, "Heat-Assisted Magnetic Recording," *IEEE Trans. on Magn.*, vol. 42, no. 10, pp. 2417–2421, Oct. 2006.
- [41] B. Terris, T. Thomson, and G. Hu, "Patterned Media for Future Magnetic Data Storage," *Microsyst. Technol.*, vol. 13, no. 2, pp. 189–196, Nov. 2006.
- [42] R. Wood, M. Williams, A. Kavcic, and J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 917–923, Feb. 2009.
- [43] Y. Chen and S. G. Srinivasa, "Joint self-iterating equalization and detection for Two-Dimensional Intersymbol-Interference," *IEEE Trans. on Communications*, vol. 61, no. 8, pp. 3219–3230, Aug. 2013.
- [44] S. G. Srinivasa, Y. Chen, and S. Dahandeh, "A communication-theoretic framework for 2-DMR channel modeling: Performance evaluation of coding and signal processing methods," *IEEE Trans. on Magn.*, vol. 50, no. 3, pp. 6–12, Mar. 2014.
- [45] N. Singla, J. O'Sullivan, R. Indeck, and Y. Wu, "Iterative decoding and equalization for 2-d recording channels," *Magnetics, IEEE Trans. on*, vol. 38, no. 5, pp. 2328–2330, Sep 2002.
- [46] Y. Wu, J. O'Sullivan, N. Singla, and R. Indeck, "Iterative detection and decoding for separable two-dimensional intersymbol interference," *Magn., IEEE Trans. on*, vol. 39, no. 4, pp. 2115–2120, July 2003.
- [47] M. Khatami and B. Vasić, "Generalized belief propagation detector for TDMR microcell model," *IEEE Trans. Magn.*, vol. 49, no. 7, pp. 3699–3702, Jul. 2013.
- [48] C. K. Matcha, S. G. Srinivasa, S. Khatami, and B. Vasić, "Two-dimensional noise-predictive maximum likelihood method for magnetic recording channels," *IEEE International Symposium on Information Theory and its Applications*, Oct. 2014.
- [49] B. Vasić, S. M. Khatami, Y. Okamoto, Y. Nakamura, Y. Kanai, J. R. Barry, S. W. McLaughlin, and E. B. Sadeghian, "A study of TDMR signal-processing opportunities based on quasi-micromagnetic simulations (invited talk)," in *Proc. The Magnetic Recording Conference (TMRC)*, Berkeley, CA, USA, August 11–13 2014, pp. 1–5.

- [50] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
- [51] J. L. Fan, T. L. Poo, and B. H. Marcus, "Constraint gain," *IEEE Trans. on Inf. Theory*, vol. 50, no. 9, pp. 1989–2001, Sept 2004.
- [52] B. Marcus, R. Roth, and P. Siegel, *Constrained systems and coding for recording channels*. Technion-I.I.T., Department of Computer Science, 1998.
- [53] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.
- [54] A. Bassalygo, "Correcting codes with an additional property," *Probl. Inf. Transm.*, vol. 4, no. 1, pp. 1–5, 1968.
- [55] M. Mansuripur, "Enumerative modulation coding with arbitrary constraints and postmodulation error correction coding for data storage systems," in *Proc. SPIE*, 1991, p. 1499.
- [56] J. C. de Souza, B. H. Marcus, R. New, and B. A. Wilson, "Constrained systems with unconstrained positions," *IEEE Trans. Inf. Theory*, vol. 48, no. 4, pp. 866–879, Apr 2002.
- [57] A. J. van Wijngaarden and K. A. S. Immink, "Efficient error control schemes for modulation and synchronization codes," in *Proceedings. IEEE Int. Symp. Inf. Theory*, Aug 1998, p. 74.
- [58] M. Khatami, M. Bahrami, and B. Vasić, "Information rates of constrained TDMR channels using generalized belief propagation," in *GLOBECOM*, San Diego, CA, Dec. 2015, pp. 1–6.
- [59] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug 2006.
- [60] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on two-dimensional constraints," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 824–838, May 2004.
- [61] R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes," *IEEE Trans. on Inf. Theory*, vol. IT-29, no. 1, pp. 5–22, Jan. 1983.
- [62] B. Marcus, P. Siegel, and J. Wolf, "Finite-state modulation codes for data storage," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 5–37, 1992.

- [63] S. W. Golomb, “Undecidability and nonperiodicity for tilings of the plane,” *Inventiones Math.*, vol. 12, pp. 177–209, 1971.
- [64] R. Berger, *The undecidability of the Domino problem*. American Mathematical Society, 1966.
- [65] R. M. Roth, P. H. Siegel, and J. K. Wolf, “Efficient coding schemes for the hard-square model,” *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1166–1176, Mar. 2001.
- [66] S. Forchhammer and T. V. Laursen, “Entropy of bit-stuffing-induced measures for two-dimensional checkerboard constraints,” *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1537–1546, Apr. 2007.
- [67] I. Tal and R. M. Roth, “Bounds on the rate of 2-D bit-stuffing encoders,” *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2561–2567, Jun. 2010.
- [68] A. Sharov and R. M. Roth, “Two dimensional constrained coding based on tiling,” *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1800–1807, Apr. 2010.
- [69] A. R. Krishnan and B. Vasić, “Lozenge tiling constrained codes,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 27, no. 4, pp. 521–542, Oct. 2014.
- [70] S. Halevy and R. M. Roth, “Parallel constrained coding with application to two-dimensional constraints,” *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1009–1020, Aug. 2002.
- [71] I. Tal, T. Etzion, and R. M. Roth, “On row-by-row coding for 2-D constraints,” *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3565–3576, Aug. 2009.
- [72] B. Vasić and K. Pedagani, “Run-length-limited low-density parity check codes based on deliberate error insertion,” *IEEE Trans. on Magn.*, vol. 40, no. 3, pp. 1738–1743, May 2004.
- [73] Z. Li, J. Xie, and B. V. K. V. Kumar, “An improved bit-flipping scheme to achieve run length control in coded systems,” *IEEE Trans. Magn.*, vol. 41, no. 10, pp. 2980–2982, Oct 2005.
- [74] Z. Li and B. V. K. V. Kumar, “Low-density parity-check codes with run length limited (RLL) constraints,” *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 344–349, Feb 2006.
- [75] H. Chou, Y. Ueng, M. Lin, and M. Fossorier, “An RLL-constrained LDPC coded recording system using deliberate flipping and flipped-bit detection,” *IEEE Trans. on Commun.*, vol. 60, no. 12, pp. 3587–3596, December 2012.

- [76] S. W. Golomb, "Tiling with sets of polyominoes," *Journal of Combinatorial Theory*, vol. 9, pp. 60–71, 1970.
- [77] J. Ashley and B. Marcus, "Two-dimensional low-pass filtering codes," *IEEE Trans. on Communications*, vol. 46, no. 6, pp. 724–727, 1998.
- [78] A. Kavcic, X. Huang, B. Vasić, W. Ryan, and M. Erden, "Channel modeling and capacity bounds for two-dimensional magnetic recording," *Magn., IEEE Trans. on*, vol. 46, no. 3, pp. 812–818, Mar. 2010.
- [79] M. Khatami, V. Ravanmehr, and B. Vasić, "GBP-based detection and symmetric information rate for rectangular-grain TDMR model," in *Proc. IEEE Int. Symp. Inf. Theory*, June 2014, pp. 1618–1622.
- [80] N. V. Abhinav Das and N. Kashyap, "MCMC methods for drawing random samples from the discrete-grains model of a magnetic medium," *IEEE Journal on Selected Areas in Commun.*, vol. 34, no. 9, pp. 2430–2438, Sep. 2016.
- [81] Kasetelyn, "Statistics of dimers on a lattice," *Physica*, vol. 27, no. 12, pp. 1209–1225, 1961.
- [82] N. Elkies, G. Kuperberg, M. Larsen, and J. Propp, "Alternating-sign matrices and domino tilings (part I)," *J. Algebraic Comb.*, vol. 1, no. 2, pp. 111–132, 1992.
- [83] S. W. Golomb, "Tiling with sets of polyominoes," *Journal of Combinatorial Theory*, vol. 9, pp. 60–71, 1970.
- [84] J. H. Conway and J. C. Lagarias, "Tiling with polyominoes and combinatorial group theory," *Journal of Combinatorial Theory Series A*, vol. 53, no. 2, pp. 183–208, 1990.
- [85] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channels," *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277–1290, Nov. 1989.
- [86] H. Pfister, J. B. Soriaga, and P. Siegel, "On the achievable information rates of finite state ISI channels," in *IEEE Glob. Telecommun. Conf. 2001*, vol. 5, 2001, pp. 2992–2996 vol.5.
- [87] P. Sadeghi, P. O. Vontobel, and R. Shams, "Optimization of information rate upper and lower bounds for channels with memory," *IEEE Trans. on Inf. Theory*, vol. 55, no. 2, pp. 663–688, Feb 2009.
- [88] G. Han, "A randomized algorithm for the capacity of finite-state channels," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3651–3669, July 2015.

- [89] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics-Doklady*, vol. 10, no. 8, Feb. 1966.
- [90] C. Liang, C. Cheng, Y. Lai, L. Chen, and H. H. Chen, “Hardware-efficient belief propagation,” *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 21, no. 5, pp. 525–537, May 2011.
- [91] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain,” in *Proc. Int. Conf. on Commun.*, vol. 2, Jun 1995, pp. 1009–1013.
- [92] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” *IEEE Trans. on Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.
- [93] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary ldpc codes over $GF(q)$,” *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 633–643, April 2007.
- [94] K. Petersen, J. Fehr, H. Burkhardt, and G. Rigoll, “Fast generalized belief propagation for map estimation on 2d and 3d grid-like markov random fields,” in *Pattern Recognition, DAGM 2008*. Springer Berlin Heidelberg, 2008.
- [95] S. Chen and Z. Wang, “Acceleration strategies in generalized belief propagation,” *IEEE Trans. on Industrial Informatics*, vol. 8, no. 1, pp. 41–48, Feb 2012.
- [96] A. D. Shigyo and K. Ishibashi, “QR-decomposed generalized belief propagation with smart message reduction for low-complexity mimo signal detection,” in *2017 Asia-Pacific Signal and Inf. Processing Assoc. Annual Summit and Conf. (APSIPA ASC)*, Dec 2017, pp. 1795–1799.
- [97] M. Jordan and C. Bishop, *An Introduction to Graphical Models*. draft, 2000.