A Deliberate Bit Flipping Coding Scheme for Data-Dependent Two-Dimensional Channels

A Deliberate Bit Flipping Coding Scheme for Data-Dependent Two-Dimensional Channels

Mohsen Bahrami, *Student Member, IEEE,* and Bane Vasić, *Fellow, IEEE*

***Abstract*—In this paper, we present a *deliberate bit flipping* (DBF) coding scheme for binary two-dimensional (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 *generalized belief propagation* algorithm is used to find an approximate solution for the problem. We evaluate 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.**

***Index Terms*—Data dependent channels, constrained coding, probabilistic inference, graphical models, and generalized belief propagation (GBP).**

I. Introduction

Recent advances in magnetic recording systems [3], [4], optical recording devices [5] and flash memory drives [6] necessitate to study *two-dimensional* (2-D) coding techniques for reliable storage/retrieval of user data. 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 [7]–[9]. As another example, in two-dimensional magnetic recording channels, 2-D isolated-bits patterns [10] 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 [11] 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 [12], *etc*. 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) constraints [13], [14] 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 [15], [16]. Designing channel codewords satisfying both ECC and channel constraints is important as it would achieve the channel capacity [17]. However, in practice this is difficult, and we rely on sub-optimal methods such as forward concatenation method (standard concatenation) [18], reverse concatenation method (modified concatenation) [19], [20], and combinations of these approaches [21], [22].

As discussed earlier, constrained codes have been used to overcome effects of harmful patterns in 1-D information storage systems. In [23], 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 [24]. 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 to the 1-D case [25], [26]). A number of *variable-rate* encoding methods have been proposed for 2-D constrained channels, including bit-stuffing encoders [10], [27]–[29] and tiling based encoders [30], [31]. Furthermore, various *row-by-row* coding methods for specific 2-D constraints were presented in [32], [33]. Vasić and Pedagani proposed an alternative approach in [34], 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

M. Bahrami and B. Vasić are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721 USA (e-mails: bahrami@email.arizona.edu and vasic@ece.arizona.edu). This work is supported by the National Science Foundation under grants ECCS-1500170 and SaTC-1813401. Parts of this paper were presented in IEEE GLOBECOM conference 2016 [1] and 8th International Conference on Algebraic Informatics (CAI 2019) [2].

of the ECC to correct both the deliberate errors and channel errors at the receiver. In [35]–[37], 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 [34] still is the number of deliberate bit errors that may overwhelm the ECC decoder and affect the error-floor performance (which limits its applications).

Our Contributions: One of the practical motivations to design a DBF coding scheme for data-dependent channels is to address the error propagation phenomena existing in conventional 2-D constrained coding methods. Most of these constrained coding 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. On the other hand, the main problem with the DBF method is the number of deliberate flips. This problem becomes 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 which may overwhelm the error-correction decoder. In this paper, 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 Generalized Belief Propagation *(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 [38], [39]. 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. 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 bits contained in any of the harmful configurations are larger than for the other bits. 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 [10] and [40], respectively.

Fig. 1. Two examples of polyominoes: (a) a 2×2 square and (b) a cross.

Paper Organization: The rest of this paper is organized as follows. Section II presents the notations and definitions used throughout the paper. In Section III, the data-dependent channel model is introduced. In Section IV, the problem of minimizing the number of flipped bits in the DBF method is formulated. In Section V, 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 VI. Section VII concludes the paper.

II. Notations and Definitions

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 $\{n_1, n_1 + k, n_1 + 2k \ldots, n_2\}$, 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. $\mathcal{A}_{m,n} = \left\{(i, j) \in \mathbb{Z}^2 : i \in [m] \text{ and } j \in [n]\right\}$ 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 $w_H(\mathbf{x}) = \sum_{x_{i,j}\in\mathbf{x}} \mathbb{1}\{x_{i,j} = 1\}$, where $\mathbb{1}\{.\}$ 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-$[N, K, d_{\text{min}}]$. A binary Reed-Muller code of length $N = 2^m$ with $N - K = 2^m - \sum_{i=0}^{r} \binom{m}{i}$ parity bits and minimum distance $d_{\text{min}} = 2^{m-r}$ is denoted by RM-(r, m).

A *polyomino* of order k, called also a k-*ominoe*, is a plane geometric figure formed by joining k neighboring square shapes. Among polyominoes are 2×2 square-shaped polyominoes

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

and cross-shaped polyominoes

$$Q^{+}(i, j) = \{(i, j - 1), (i - 1, j), (i, j), (i, j + 1), (i + 1, j)\}\,, \quad (2)$$

over the 2-D lattice $\mathbb{Z}^2$, which are shown in Fig. 1.

An $m \times n$ binary pattern is denoted by $\mathbf{x} = [x_{i,j}]_{i \in [m], j \in [n]}$, where $x_{i,j}$ indicates the value of bit in i-th row and j-th column. Throughout the paper, white squares denote zero bits and black squares represent 1. 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 $\mathbf{x}_\mathcal{P}$. As an example, Fig. 2 shows all binary configurations of a 2×2 square-shaped polyomino.

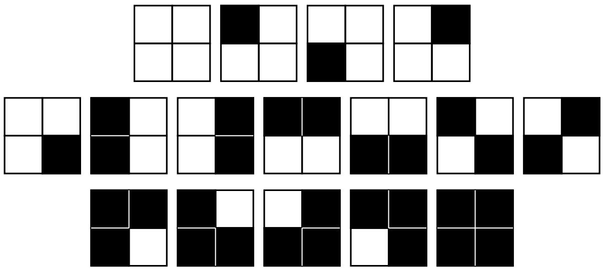


Fig. 2. The set of all binary configurations of a 2×2 square-shaped polyomino.

Consider $x_{i,j}$ over an $m \times n$ rectangular pattern $\mathbf{x}$, then the union of all $\mathcal{P}$-shaped polyominoes that intersect with this bit 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

$$\mathcal{P}_{i,j}^{\square} = \bigcup_{(i',j') \in Q^{\square}(i-1,j-1)} Q^{\square}(i',j'), \tag{3}$$

and

$$\mathcal{P}_{i,j}^{+} = \bigcup_{(i',j') \in Q^{+}(i,j)} Q^{+}(i',j'), \tag{4}$$

respectively. Fig. 3 shows $\mathcal{P}_{i,j}$ for these polyominoes.

III. Channel Model

In this section, we introduce a communication channel transmitting binary rectangular patterns and producing as an output a binary pattern. 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 bits contained in configurations which belong to the set of harmful configurations is larger than the other bits. Therefore, the channel has states and its error statistics depends on input binary patterns. In the following, we formally present error and state characterizations.

(a) (b)

Fig. 3. Figure demonstrates $\mathcal{P}_{i,j}$ over a rectangle when the polyomino is: (a) a 2×2 square and (b) a cross.

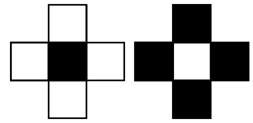


Fig. 4. 2-D isolated-bits patterns containing the bit $x_{i,j}$.

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 $x_{i,j}$, $\mathcal{P}_{i,j}$ denotes the union of $\mathcal{P}$-shaped polyominoes that intersect with this bit, and $\mathbf{x}_{\mathcal{P}_{i,j}}$ is the configuration of $\mathcal{P}_{i,j}$, as defined in Section II. 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 $\mathcal{X}^{G}_{\mathcal{P}_{i,j}}$ and $\mathcal{X}^{B}_{\mathcal{P}_{i,j}}$, i.e., $\mathcal{X}_{\mathcal{P}_{i,j}} = \mathcal{X}^{G}_{\mathcal{P}_{i,j}} \bigcup \mathcal{X}^{B}_{\mathcal{P}_{i,j}}$, where $\mathcal{X}^{B}_{\mathcal{P}_{i,j}}$ is the set of configurations containing $\mathcal{P}$-shaped configurations which are harmful for the channel. For example, $\mathcal{X}^{B}_{\mathcal{P}_{i,j}}$ can be the set of binary configurations of $\mathcal{P}_{i,j}$ given in Fig. 3(b), which contains the 2-D *isolated-bit* patterns. The 2-D isolated-bit patterns are shown in Fig. 4.

For $x_{i,j}$ 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 though the channel, a bit 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 bits contained in a harmful configuration is much larger than that of the other bits. The received binary pattern is $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}^{\text{CH}}$, where $\mathbf{e}^{\text{CH}} = [e^{\text{CH}}_{i,j}]$ is the channel error array. Therefore, $e^{\text{CH}}_{i,j}$ 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 $\mathbf{x}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}^{B}_{i,j}}$ and a BSC with crossover probability α_g when $\mathbf{x}_{\mathcal{P}_{i,j}} \notin \mathcal{X}_{\mathcal{P}^{B}_{i,j}}$, respectively.

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

$$f_{\text{CH}}\left(\mathbf{x}_{\mathcal{P}_{i,j}}\right) = \mathbb{1}\left\{\mathbf{x}_{\mathcal{P}_{i,j}} \in \mathcal{X}_{\mathcal{P}^{B}_{i,j}}\right\}, \tag{5}$$

to identify bits which are contained in harmful configurations, where $x_{i,j}$ belongs to at least one harmful configuration if $f_{\text{CH}}\left(\mathbf{x}_{\mathcal{P}_{i,j}}\right) = 1$. Using the above indicator function, we can determine the channel state for transmission of $x_{i,j}$ as follows

$$s_{i,j} = \begin{cases} b, & f_{\text{CH}}\left(\mathbf{x}_{\mathcal{P}_{i,j}}\right) = 1, \\ g, & f_{\text{CH}}\left(\mathbf{x}_{\mathcal{P}_{i,j}}\right) = 0, \end{cases} \tag{6}$$

where "b" and "g" stand for the bad and the good channel states, respectively. Let the probability distribution function of channel be $p(\mathbf{y}|\mathbf{x})$. According to the aforementioned error characterization, the probability distribution function of channel can be factored into

$$p(\mathbf{y}|\mathbf{x}) = \prod_{(i,j)} p\left(y_{i,j} | \mathbf{x}_{\mathcal{P}_{i,j}}\right), \tag{7}$$

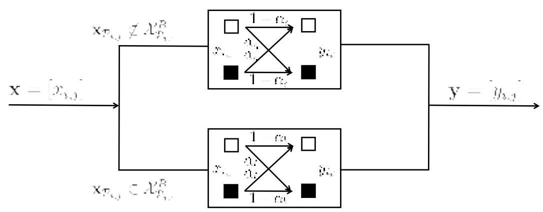


Fig. 5. A schematic representation for the channel model is given. Passing through the channel, $x_{i,j}$ is in error with probability α_b if the configuration of $\mathcal{P}_{i,j}$, $\mathbf{x}_{\mathcal{P}_{i,j}}$, belongs to the set of harmful patterns $\mathcal{X}^B_{\mathcal{P}_{i,j}}$, otherwise it inverts with a probability of α_g. It should be noted that the top arm of the figure can be removed when $\alpha_g = 0$, which reduces the channel into a constrained 2-D channel with the list of forbidden configurations $\mathcal{X}^B_{\mathcal{P}_{i,j}}$. However, in our channel removing the harmful patterns does not make the channel noiseless. Removing all the harmful patterns in the set $\mathcal{X}^B_{\mathcal{P}_{i,j}}$ before transmission through the channel, makes it a BSC with the cross-over probability α_g.

since $y_{i,j}$ only depends on the configuration of $\mathcal{P}_{i,j}$ in the input pattern $\mathbf{x}$. Fig. 5 gives a schematic illustration for the channel.

Remark 1: In this paper, we use the concept of polyominoes to just demonstrate the effect of harmful configurations on its neighboring bits over a 2-D binary pattern. As two examples, we consider 4-ominoes and 5-ominoes, as these reflect physical effects of 2-D ISI and ICI over the plane. For this purpose, we defined the square and cross shaped polyominoes in (1) and (2).

Remark 2: The channel is similar to the Gilbert-Elliot channel [41], as it has two states, where 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 [42]–[44].

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 $x_{i,j}$ is

$$p\left(s_{i,j} = b\right) = p\left(f_{\text{CH}}\left(\mathbf{x}_{\mathcal{P}_{i,j}}\right) = 1\right) = \frac{|\mathcal{X}^B_{\mathcal{P}_{i,j}}|}{|\mathcal{X}_{\mathcal{P}_{i,j}}|}, \tag{8}$$

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

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 binary patterns passing through the channel.

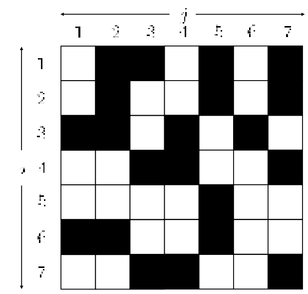


Fig. 6. A 7×7 binary pattern $\mathbf{x}$ is transmitted through the channel with the set of 2-D isolated-bits patterns as the set of harmful patterns. The bits $x_{2,6}$, $x_{3,5}$, $x_{3,6}$, $x_{3,7}$, $x_{4,6}$, $x_{6,7}$, $x_{7,6}$ and $x_{7,7}$ belong to the 2-D isolated-bits patterns. Passing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

Example 1: Fig. 6 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, which are given in Fig. 4. In order to determine the channel state for all bits over the pattern, we assume zero entries 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 bits contained in these two harmful configurations are in error with a probability of α_b. These bits $x_{2,6}$, $x_{3,5}$, $x_{3,6}$, $x_{3,7}$, $x_{4,6}$, $x_{6,7}$, $x_{7,6}$ and $x_{7,7}$. For instance, for $x_{2,6}$,

$$\mathcal{P}_{2,6} = \bigcup_{(i',j') \in Q^+(2,6)} Q^+(i', j'). \tag{9}$$

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, $x_{2,6}$ is in the bad state. Similarly, we can check this for the rest of bits in $\mathbf{x}$.

IV. Problem Formulation

The user uniformly and randomly selects a binary message $\mathbf{m}$ out of 2^K messages denoted by $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{2^K}\}$, 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 $\phi_{\text{ECC}} : \mathcal{M} \to \mathcal{S}^N_{\text{ECC}}$ assigns a binary codeword $\mathbf{c}(\mathbf{m})$ of length N to the user data $\mathbf{m}$ such that

$$\mathbf{c}(\mathbf{m}) = \phi_{\text{ECC}}(\mathbf{m}), \tag{10}$$

where $\mathcal{S}^N_{\text{ECC}} = \{\mathbf{c}(\mathbf{m}_1), \mathbf{c}(\mathbf{m}_2), \ldots, \mathbf{c}(\mathbf{m}_{2^{\lfloor NR \rfloor}})\}$ is the codebook (the set of binary codewords of length N) associated with the ECC being used. A codeword $\mathbf{c} \in \mathcal{S}^N_{\text{ECC}}$ is represented by N binary symbols, $\mathbf{c} = (c_1, c_2, \ldots, c_N)$, 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 III, with the list of harmful configurations $\mathcal{X}^B_{\mathcal{P}_{i,j}}$. Assuming that $\alpha_b \gg \alpha_g$, then bits contained in configurations of list $\mathcal{X}^B_{\mathcal{P}_{i,j}}$ are more prone to error than the other bits. 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 $\mathcal{X}^B_{\mathcal{P}_{i,j}}$ in the input pattern $\mathbf{x}$, the color of selected bits 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 bits $x_{3,6}$ and $x_{7,7}$.

This method of eliminating harmful configurations from binary patterns with deliberating flipping bits 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

$$\hat{\mathbf{x}} = \phi(\mathbf{x}). \tag{11}$$

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

$$\mathbf{e}^{\text{DBF}} = \theta(\mathbf{x}) = [e^{\text{DBF}}_{i,j}]_{i \in [m], j \in [n]}, \tag{12}$$

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

$$\phi(\mathbf{x}) = \mathbf{x} \oplus \theta(\mathbf{x}), \tag{13}$$

and the number of flipped bits 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 $\mathbf{e}_{\text{CH}}$ 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

$$\mathbf{y} = \mathbf{x} \oplus \mathbf{e}^{\text{CH}} \oplus \mathbf{e}^{\text{DBF}}. \tag{14}$$

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

$$\begin{aligned} &\psi : \mathcal{Y} \rightarrow \mathcal{M}, \\ &\hat{\mathbf{m}} = \psi(\mathbf{y}). \end{aligned} \tag{15}$$

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

$$p_e^{(N)} = p(\hat{\mathbf{m}} \neq \mathbf{m}) = \sum_{\mathbf{m} \in \mathcal{M}} \lambda_{\mathbf{m}} p(\mathbf{m}) \overset{(a)}{=} \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} \lambda_{\mathbf{m}}, \tag{16}$$

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

$$d_H\left(\mathbf{x}(\mathbf{m}), \mathbf{y}\right) \leq \lfloor \frac{d_{\min} - 1}{2} \rfloor, \tag{17}$$

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 $w_H\left(\mathbf{e}^{\text{CH}}\right) \simeq 0$ after removing harmful patterns from the input pattern. Under this assumption, we have $\mathbf{y} \simeq \mathbf{x} \oplus \mathbf{e}^{\text{DBF}}$ and

$$d_H\left(\mathbf{x}, \mathbf{y}\right) \simeq d_H\left(\mathbf{x}, \mathbf{x} \oplus \mathbf{e}^{\text{DBF}}\right) = w_H(\mathbf{e}^{\text{DBF}}). \tag{18}$$

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

$$\lambda_{\mathbf{m}} = p(\hat{\mathbf{m}} \neq \mathbf{m} \mid \mathbf{m}) \simeq p\left(w_H(\mathbf{e}^{\text{DBF}}) > \lfloor \frac{d_{\min} - 1}{2} \rfloor \middle| \mathbf{m}\right), \tag{19}$$

and

$$p_e^{(N)} \simeq \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p\left(w_H(\mathbf{e}^{\text{DBF}}) > \lfloor \frac{d_{\min} - 1}{2} \rfloor \middle| \mathbf{m}\right), \tag{20}$$

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 [11], "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 (20). There may be different choices of deliberate errors $\mathbf{e}^{\text{DBF}}$ 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 flipping bits more than the number of errors that the decoder can correct. Ideally, the bit 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 those 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, $w_H\left(\mathbf{e}^{\text{DBF}}\right)$ should be minimized for each message $\mathbf{m} \in \mathcal{M}$. Therefore, the roles of the bit selection function θ are *(i)* to identify and remove the harmful configurations $\mathcal{X}^B_{\mathcal{P}_{i,j}}$ 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 bit 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.,

$$\mathcal{E}^{\mathbf{x}} = \left\{\mathbf{e}^{\text{DBF}} | \hat{\mathbf{x}} = \mathbf{x} \oplus \mathbf{e}^{\text{DBF}} \in \mathbb{S}\right\}. \tag{21}$$

In order to minimize the average probability of error in (20), we need to find an error pattern $\mathbf{e}^{\star}_{\text{DBF}}$ which has the minimum Hamming weight among the error patterns in $\mathcal{E}^{\mathbf{x}}$, or another word,

$$\mathbf{e}^{\star}_{\text{DBF}} = \arg \min_{\mathbf{e}^{\text{DBF}} \in \mathcal{E}^{\mathbf{x}}} \left\{w_H(\mathbf{e}^{\text{DBF}})\right\}. \tag{22}$$

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

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 $\mathbf{c}_1 = (0,1,0,0,0,1,1,1,1,0,1,0,1,1,0)$, $\mathbf{c}_2 = (1,0,0,0,0,1,0,1,0,0,1,1,0,1,1)$, $\mathbf{c}_3 = (0,1,1,1,1,0,1,0,1,1,0,0,1,0,0)$, and $\mathbf{c}_4 = (0,1,1,0,1,1,1,0,0,0,0,1,0,1,0)$ of length 15 which are then arranged into 3×5 arrays as four different patterns. The patterns are shown in Fig. 7, 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 flipping scenarios using the deliberate error insertion method.

We are interested in removing 2-D isolated-bits configurations entirely from the above patterns with minimal number of bit flips. In other words, the goal is to find the error pattern $\mathbf{e}^{\text{DBF}}$ for each input pattern $\mathbf{x}$ which has the minimum

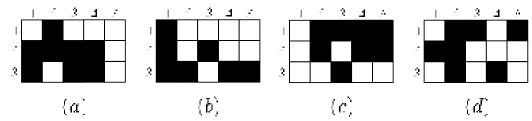


Fig. 7. The input patterns for Example 2. We assume zero entries outside of each input pattern.

Hamming weight and $\mathbf{x} \oplus \mathbf{e}^{\text{DBF}}$ does not contain any of the 2-D isolated-bits configurations. Therefore, we have

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

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

In Fig. 7(a), the pattern does not contain any of the 2-D isolated-bits configurations, therefore there is no need to flip any bit, and $w_H(\mathbf{e}_{(a)}) = 0$. The pattern in Fig. 7(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 bits in $Q^+(2,3)$, and therefore $w_H(\mathbf{e}_{(b)}) = 1$. For the pattern in Fig. 7(c), there are two overlapping 2-D isolated-bits patterns, which are $\mathbf{x}_{Q^+(2,3)}$ and $\mathbf{x}_{Q^+(3,3)}$. These two isolated-bits patterns can be removed simultaneously by flipping either $x_{2,3}$ or $x_{3,3}$, and therefore for this case also $w_H(\mathbf{e}_{(c)}) = 1$. In Fig. 7(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 flip at least two bits over this input pattern, and for this case $w_H(\mathbf{e}_{(d)}) = 2$. For the above systematic BCH-$[15,5,7]$ code (where the codewords are arranged into 3×5 arrays and the first row is equipped with the user bits), we identified the minimum number of bit flips required for removing 2-D isolated bit patterns from each of the possible BCH-$[15,5,7]$ codewords. Assuming the codewords are chosen randomly and uniformly, in average it needs to flip 0.6563 bits/pattern to remove the forbidden configurations by the 2-D n.i.b. constraint from an input pattern.

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 [45], 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 $\mathbb{S}$. The capacity of 2-D constraint is given by

$$C_{\text{2-D}} = \lim_{m,n\to\infty} \frac{1}{m\times n}\log_2 Z(m,n), \tag{23}$$

where $Z(m,n)$ is the number of admissible $m\times n$ binary patterns, i.e.,

$$Z(m,n) = \left|\mathbb{S}\cap\{0,1\}^{m\times n}\right|. \tag{24}$$

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

$$p(\hat{\mathbf{x}}) = \begin{cases} \frac{1}{|\mathbb{S}|}, & \hat{\mathbf{x}}\in\mathbb{S},\\ 0, & \text{other}.\end{cases} \tag{25}$$

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.

V. A Probabilistic Graphical Formultion for Minimzing 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 replaced with a 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 a large pattern. As it can be verified for $x_{i,j}$ locally over a finite neighborhood of bits $\mathcal{P}_{i,j}$ whether the bit is contained in a harmful pattern of the set $\mathcal{X}^B_{\mathcal{P}_{i,j}}$, we define a *local distortion* function D for each $x_{i,j}$ 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 $x_{i,j}\in\mathcal{A}_{m,n}$, the function $D:\{0,1\}^{|\mathcal{P}_{i,j}|}\times\{0,1\}^{|\mathcal{P}_{i,j}|}\to\mathbb{N}$ is defined over $x_{\mathcal{P}_{i,j}}$ as follows

$$D\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}},\mathbf{x}_{\mathcal{P}_{i,j}}\right) = \begin{cases} w_H\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\oplus\mathbf{x}_{\mathcal{P}_{i,j}}\right), & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\notin\mathcal{X}^B_{\mathcal{P}_{i,j}},\\ \infty, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}},\end{cases} \tag{26}$$

where $w_H\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\oplus\mathbf{x}_{\mathcal{P}_{i,j}}\right)$ 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 $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\notin\mathcal{X}^B_{\mathcal{P}_{i,j}}$ which have the same Hamming distance with $\mathbf{x}_{i,j}$. One may use the outputs of D for the bits $x_{i,j}\in\mathcal{A}_{m,n}$ to find $\mathbf{x}^\star\in\mathbb{S}$ 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 bit $x_{i,j}\in\mathcal{A}_{m,n}$, 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 (26), we use the binomial expression to translate the Hamming distance metric into the probability domain. We assume that the color of each bit contained in a harmful configuration is inverted with the probability $0<\lambda\le 1$. For every bit $x_{i,j}\in\mathcal{A}_{m,n}$, we define a function $D_p:\{0,1\}^{\mathcal{P}_{i,j}}\times\{0,1\}^{\mathcal{P}_{i,j}}\to\mathbb{R}^{[0,1]}$ over the bits indexed by $\mathcal{P}_{i,j}$,

$$\begin{aligned} &D_p(\mathbf{x}_{\mathcal{P}_{i,j}},\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}) =\\ &\quad\begin{cases} \lambda^{w_H(\mathbf{e}_{\mathcal{P}_{i,j}})}(1-\lambda)^{|\mathcal{P}_{i,j}|-w_H(\mathbf{e}_{\mathcal{P}_{i,j}})}, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\notin\mathcal{X}^B_{\mathcal{P}_{i,j}},\\ 0, & \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}},\end{cases}\end{aligned} \tag{27}$$

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 bits in $\mathcal{P}_{i,j}$. This function is called as the *local probabilistic distortion* function. For each bit $x_{i,j}\in\mathcal{A}_{m,n}$, 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 $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\notin\mathcal{X}^B_{\mathcal{P}_{i,j}}$. When $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}}$, 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 $\mathcal{X}^B_{\mathcal{P}_{i,j}}$, we are now interested in finding $\hat{\mathbf{x}}\in\mathbb{S}$ maximizing $p\left(\hat{\mathbf{x}}|\mathbf{x}\right)$, which is equivalent to finding $\hat{\mathbf{x}}$ that minimizes $w_H\left(\hat{\mathbf{x}}\oplus\mathbf{x}\right)$. In another word, we want to find

$$\hat{\mathbf{x}} = \arg\max_{\hat{\mathbf{x}}\in\mathbb{S}}\left\{p(\hat{\mathbf{x}}|\mathbf{x})\right\}. \tag{28}$$

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

$$\begin{aligned} &\max_{\hat{\mathbf{x}}\in\mathbb{S}} p\left(\hat{\mathbf{x}}|\mathbf{x}\right) = \max_{\hat{\mathbf{x}}\in\mathbb{S}}\frac{p\left(\mathbf{x}|\hat{\mathbf{x}}\right)p\left(\hat{\mathbf{x}}\right)}{p\left(\mathbf{x}\right)} \overset{(a)}{\propto} \max_{\hat{\mathbf{x}}\in\mathbb{S}} p\left(\mathbf{x}|\hat{\mathbf{x}}\right)\\ &\overset{(b)}{=} \max_{\hat{\mathbf{x}}\in\mathbb{S}} \prod_{(i,j)\in\mathcal{A}_{m,n}} p\left(x_{i,j}|\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\right),\\ &\overset{(c)}{=} \max_{\hat{\mathbf{x}}\in\mathbb{S}} \prod_{(i,j)\in\mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}}\}}(1-\lambda)^{1-\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}}\}},\end{aligned} \tag{29}$$

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 $x_{i,j}$ we can determine locally over $\mathcal{P}_{i,j}$ that the bit is contained in a harmful pattern, and (c) is obtained based on the definition of the local probabilistic distortion function, given in (27). The probability $p\left(\mathbf{x}_{i,j} \middle| \hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\right)$ indicates that the bit $x_{i,j}$ is flipped depending on the realization of its neighboring bits $\mathbf{x}_{\mathcal{P}_{i,j}}$, whether belongs to the set of harmful configurations or not. Therefore, we have

$$p\left(\hat{\mathbf{x}}|\mathbf{x}\right) = \frac{1}{Z(\mathbf{x})} \prod_{(i,j)\in\mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}} (1-\lambda)^{1-\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}}, \tag{30}$$

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

$$Z(\mathbf{x}) = \sum_{\hat{\mathbf{x}}\in\{0,1\}^{m\times n}} \prod_{(i,j)\in\mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}} (1-\lambda)^{1-\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}}. \tag{31}$$

In order to compute the *a-posteriori* probability $p\left(\hat{\mathbf{x}}|\mathbf{x}\right)$ with the factorization given in (30), we need to calculate the partition function given in the equation (31). Providing either exact or approximate solutions for the partition function in general is a NP-hard problem [46]. In [39] and [47], 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 $p\left(\hat{\mathbf{x}}|\mathbf{x}\right)$ in (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 (28).

Remark 7: For a given input pattern $\mathbf{x}$, we should note that the zero probability in (27) ensures that an approximate solution $\hat{\mathbf{x}}$ does not contain any harmful configurations, i.e., $\hat{\mathbf{x}} \in \mathbb{S}$. However, due to the fact that the RBA method only provides an approximate solution for (28), the 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 [48]–[52]. 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.

VI. 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 (27). To illustrate the usefulness of DBF method, we investigate its performance over the data-dependent channel in Section III 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. Finally, 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 [40] and [10] respectively.

Remark 9: It should be noted that the parent-to-child message passing steps ([39]) 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) arrays 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.

A. 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. 8, 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, 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. We extend the same analysis for random input patterns of size 8×8 and obtained the average number flipped bits of 1.46 bits/pattern for removing the 2-D isolated bit patterns entirely from the input patterns.

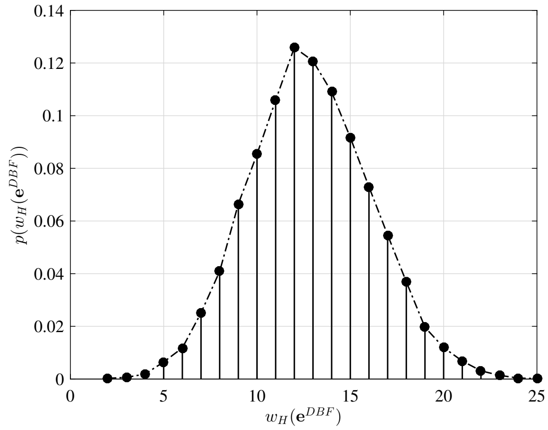


Fig. 8. An approximation of the occurrence probability of bit flipping for removing the forbidden patterns by the 2-D n.i.b. constraint from random 32×32 arrays are given over 8000 trials. For this experiment, $\lambda = 0.1$ in (28).

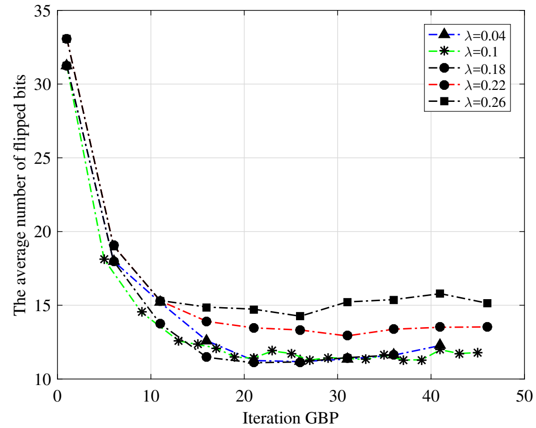


Fig. 10. The average number of flipped bits for removing 2-D isolated-bits patterns from a random 32×32 array for different $\lambda \in \{0.04, 0.1, 0.18, 0.22, 0.26\}$ over 1000 trials versus the number of GBP iterations.

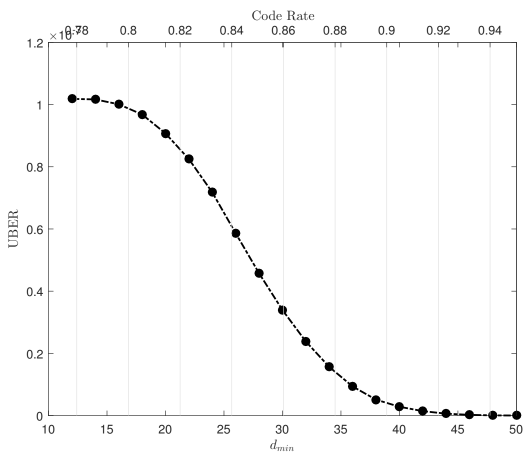


Fig. 9. BCH codes of length 1024 with different code rates are used to correct the deliberate errors introduced in random 32×32 patterns for removing 2-D isolated-bits patterns. Using the flipping probabilities in Fig. 8 and (32), the UBER is calculated for BCH codes of length 1024 with different rates (and consequently d_{min}).

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

$$\text{UBER} = \left[\sum_{w_H(\mathbf{e}^{\text{DBF}}) > \lfloor \frac{d_{\text{min}}-1}{2} \rfloor} w_H(\mathbf{e}^{\text{DBF}}) p\left(w_H(\mathbf{e}^{\text{DBF}})\right) \right] / N, \tag{32}$$

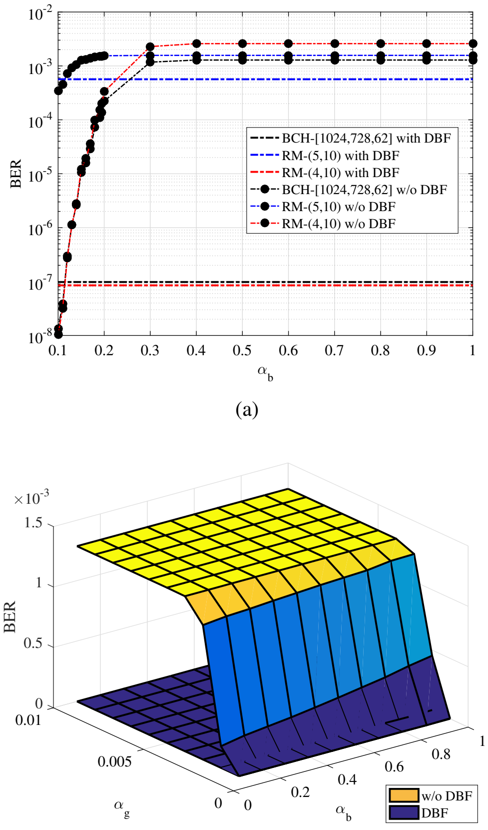
where d_{min} is the minimum distance of code, $N = m \times n$ is the size of the pattern (length of the code), and R is the rate of the ECC. Using BCH codes of length 1024 for correcting deliberate errors introduced in random 32×32 binary patterns for removing the 2-D isolated-bits configurations, the UBER is given as a function of d_{min} in Fig. 9. 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, (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. 10 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.

B. 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 III 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 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



(a)

(b)

Fig. 11. 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. The BCH-$[1024, 728, 62]$, RM-$(4, 10)$ and RM-$(5, 10)$ codes are 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.

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

$$p_e^{(\mathrm{DBF})} = \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p\left(w_H(\mathbf{e}^{\mathrm{DBF}} \oplus \mathbf{e}^{\mathrm{CH}}) > \lfloor \frac{d_{\mathrm{min}} - 1}{2} \rfloor | \mathbf{m} \right), \tag{33}$$

and

$$p_e^{(\mathrm{w/o\ DBF})} = \frac{1}{2^{\lfloor NR \rfloor}} \sum_{\mathbf{m}} p\left(w_H(\hat{\mathbf{e}}^{\mathrm{CH}}) > \lfloor \frac{d_{\mathrm{min}} - 1}{2} \rfloor | \mathbf{m} \right), \tag{34}$$

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

In Fig. 11(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 bits 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. 11(a) shows the BER results with and without incorporating DBF for different values of α_b, when the BCH-$[1024, 728, 62]$, RM-$(4, 10)$ and RM-$(5, 10)$ codes are 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 using the BCH-$[1024, 728, 62]$ code. 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. 11(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, and using the BCH-$[1024, 728, 62]$ code.

C. 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 [40]: 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 [10]: 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

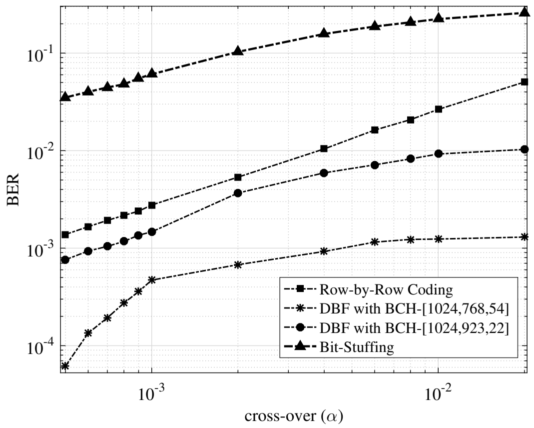


Fig. 12. Figure shows the BER comparison results of the DBF, bit-stuffing and row-by-row coding methods on the BSC with the cross-over probability (α). The effect of error propagation can be observed in the BER curve of bit-stuffing which shows that this method is vulnerable to channel errors. The coding 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.

can apply the reverse transformation to recover the original information bits. Fig. 12 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. As another comparison, we used a column-weight 4 quasi cyclic LDPC code over two-dimensional magnetic recording channels for removing harmful patterns in our earlier work [1], where an order of magnitude gain in the frame-error-rate was obtained for Voronoi based 2-D magnetic recording channels with low magnetic grain densities.

VII. Conclusions and Future Work

To summarize, we proposed 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 paper, we have focused on minimizing the number of deliberate errors in the DBF scheme for removing a set of given configurations from input patterns. We devised a probabilistic graphical model for the minimization problem by reformulating it as a 2-D MAP problem. We used 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 showed that how these numbers are comparable with the error correction capability of BCH codes being used. Furthermore, we investigated the suitability of DBF method for imposing 2-D constraint over a BSC against classical constrained coding methods which suffer from error propagation.

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 (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.

Appendix A
(The Region based Approximation Method and GBP)

In this appendix, we present factor graph and region graph representations for the constrained maximization problem given in (28). Furthermore, we explain the RBA method using GBP [39] to find an approximate solution for the problem.

For the maximization problem in (28), we showed in (29) that the *a-posteriori* probability $p\left(\hat{\mathbf{x}}|\mathbf{x}\right)$ is proportional to

$$p\left(\hat{\mathbf{x}}|\mathbf{x}\right) \propto \prod_{(i,j)\in\mathcal{A}_{m,n}} \lambda^{\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}}\}}(1-\lambda)^{1-\mathbb{1}\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\in\mathcal{X}^B_{\mathcal{P}_{i,j}}\}}. \tag{35}$$

We consider a *multiplicative factor graph* [38], i.e., a factor graph where the global function is a product of local functions. We consider a bipartite graph $\mathcal{G} = (\mathbf{X}, \mathbf{F}, \mathbf{E})$ with two sets of nodes $\mathbf{X}$ and $\mathbf{F}$, and a set of edges $\mathbf{E}$ connecting only different node types. The set $\mathbf{X}$ consists of N random variables which present the N bits over the $m \times n$ input pattern $\mathbf{x}$, where $N = m \times n$. Therefore, $\mathbf{X} = \{X_{i,j} : (i,j) \in \mathcal{A}_{m,n}\}$, and $X_{i,j}$ takes value 0 or 1. The set $\mathbf{F} = \{f_{i,j} : (i,j) \in \mathcal{A}_{m,n}\}$, and the factor node $f_{i,j}$ represents the local probabilistic distortion function $D_p(\mathbf{x}_{\mathcal{P}_{i,j}}, \hat{\mathbf{x}}_{\mathcal{P}_{i,j}})$ which is defined in (27). The factor node $f_{i,j} \in \mathbf{F}$ is connected to the variable node $X_{i,j} \in \mathbf{X}$ if the local function associated with the factor node $f_{i,j}$ involves $X_{i,j}$. This graphical model serves as a basis for the RBA scheme to solve our constrained maximization problem given in (28).

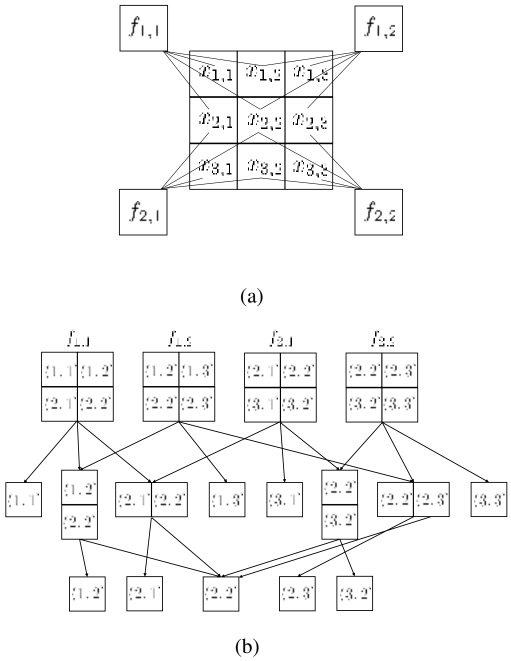


Fig. 13. (a) Factors $f_{i,j}(\cdot)$ of a 2×2 pattern are shown. (b) The region graph corresponding to the factor graph is given.

The free energy F_H is defined by $-\ln Z$ (log partition function) in statistical mechanics. Using the properties of Kullback-Liebler divergence [53], we can obtain an approximation for the free energy by minimizing the variational free energy with respect to a trial probability distribution $b(\hat{\mathbf{x}})$ for the *a-posteriori* probability distribution $p(\hat{\mathbf{x}}|\mathbf{x})$. The trial probability distribution $b(\hat{\mathbf{x}})$ should be normalized and $0 \leq b(\hat{\mathbf{x}}) \leq 1$ for all $\hat{\mathbf{x}}$. We can also consider $b(\hat{\mathbf{x}})$ as the belief of the *a-posterior* probability of $\hat{\mathbf{x}}$. The variational free energy corresponding to $b(\hat{\mathbf{x}})$ is defined by

$$F(b(\hat{\mathbf{x}})) = U(b(\hat{\mathbf{x}})) - H(b(\hat{\mathbf{x}})), \tag{36}$$

where for our problem

$$U(b(\hat{\mathbf{x}})) = - \sum_{(i,j)\in\mathcal{A}_{m,n}} \sum_{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}} b\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\right) \lambda^{1\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}} (1-\lambda)^{1-1\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}},$$

$$H(b(\hat{\mathbf{x}})) = -\sum_{\hat{\mathbf{x}}} b(\hat{\mathbf{x}}) \ln b(\hat{\mathbf{x}}), \tag{37}$$

are the average energy and entropy, respectively, and $b\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\right)$ is the corresponding belief of bits $\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}$. The variational free energy can be estimated using the RBA scheme [39], [47]. In order to use the RBA method, we need to construct a valid region graph in such a way that each variable/factor node contains at least in one region. A region graph consists of clusters of variable and factor nodes, and can be constructed from a factor graph. A region graph initially is formed by clustering every factor node and its neighboring variables nodes into a region, which is called an ancestor region, so that every ancestor region consists of one factor node and its neighboring variable nodes. Then, the cluster variation method [39] is applied to establish the remaining of the region graph. The remaining regions are formed by taking the intersection of the basic regions and their intersections – as shown in Fig. 13(b). For the region R, we denote the set of variable nodes in the region R by $\mathbf{X}_R$ and the state of these variables by $\mathbf{x}_R$. Let $b(\mathbf{x}_R)$ and $p(\mathbf{x}_R)$ be the belief and the probability of $\mathbf{x}_R$. Furthermore, we denote the collection of all the regions in the region graph by $\mathcal{R}$.

According to [39], the variational free energy can be estimated using the RBA method such that

$$\hat{F}(b(\hat{\mathbf{x}})) = U_{\mathcal{R}}(b(\hat{\mathbf{x}})) - H_{\mathcal{R}}(b(\hat{\mathbf{x}})), \tag{38}$$

where $U_{\mathcal{R}}$ and $H_{\mathcal{R}}$ are respectively the region average energy and region entropy and given by

$$U_{\mathcal{R}}(b(\hat{\mathbf{x}})) = - \sum_{i,j\in\mathcal{A}_{m,n}} \sum_{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}} b\left(\hat{\mathbf{x}}_{\mathcal{P}_{i,j}}\right) \lambda^{1\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}} (1-\lambda)^{1-1\{\hat{\mathbf{x}}_{\mathcal{P}_{i,j}} \in \mathcal{X}^B_{\mathcal{P}_{i,j}}\}},$$

$$H_{\mathcal{R}}(b(\hat{\mathbf{x}})) = \sum_{R\in\mathcal{R}} c_R \sum_{\hat{\mathbf{x}}_R} b(\hat{\mathbf{x}}_R) \ln b(\hat{\mathbf{x}}_R), \tag{39}$$

where $\hat{\mathbf{x}}_R$ are the variables belonging to the region $R \in \mathcal{R}$ and c_R is the counting number of the region R given by $c_R = 1 - \sum_{p\in\mathcal{A}_R} c_p$ where $\mathcal{A}_R$ is the set of ancestors of region R identified by $\mathcal{A}_R = \{R' \in \mathcal{R} | R \subset R'\}$. The *a-posteriori* probability $p(\hat{\mathbf{x}}|\mathbf{x})$ can be now estimated by minimizing (38) subject to the edge constraints given by

$$\sum_{\hat{\mathbf{x}}_U \in \hat{\mathbf{x}}_{P\setminus R}} b(\hat{\mathbf{x}}_U) = b(\hat{\mathbf{x}}_R) \quad \forall p \in \mathcal{P}_R, \forall R \in \mathcal{R}, \tag{40}$$

where $\hat{\mathbf{x}}_{P\setminus R}$ denotes the set of variables in the parent region P, but not in R. Furthermore, the normalization constraints are $\sum_{\hat{\mathbf{x}}_R} b(\hat{\mathbf{x}}_R) = 1, \forall R \in \mathcal{R}$. The edge constraints ensure that the belief of a region can be obtained from its parent regions. The message and belief update equations in the GBP algorithm for finding an approximate solution for the problem of minimizing the number of flipped bits in DBF method can be obtained from solving the constrained minimization problem of $\hat{F}$, given in (38), using the Lagrange multipliers.

In the considered formulation of the GBP algorithm (parent-to-child algorithm [39]) for minimizing the number of flipped bits, the size of the regions is dictated by $|\mathcal{P}_{i,j}|$. The computational complexity associated with each edge in the in this implementation is $\mathcal{O}\left(2^{|\mathcal{P}_{i,j}|}\right)$ – the proof of this analysis for the general parent-to-child GBP algorithm can be found in [54, Lemma 1]. Since the number of beliefs for each region depends on its size, it is practically not feasible to use this formulation of the algorithm when $|\mathcal{P}_{i,j}|$ is large.

References

[1] M. Bahrami and B. Vasić, "Generalized belief propagation based deliberate bit flipping modulation coding," in *Proc. IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–6.

[2] ——, "Constraint satisfaction through GBP-guided deliberate bit flipping," in *8th International Conference on Algebraic Informatics (CAI 2019)*, Nis, Serbia, June 30–July 4 2019, pp. 1–12.

[3] 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.

[4] S. S. Garani, L. Dolecek, J. Barry, F. Sala, and B. Vasić, "Signal processing and coding techniques for 2-D magnetic recording: An overview," *Proc. of the IEEE*, vol. 106, no. 2, pp. 286–318, Feb. 2018.

[5] K. A. S. Immink, "A survey of codes for optical disk recording," *IEEE J. on Selected Areas in Commun.*, vol. 19, no. 4, pp. 756–764, Apr. 2001.

[6] L. Dolecek and Y. Cassuto, "Channel coding for nonvolatile memory technologies: Theoretical advances and practical considerations," *Proc. of the IEEE*, vol. 105, no. 9, pp. 1705–1724, Sept. 2017.

[7] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 2128–2132.

[8] ——, "Low-complexity two-dimensional data encoding for memory inter-cell interference reduction," in *Proc. 27th Conv. IEEE Israel (IEEEI)*, Nov. 2012, pp. 1–5.

[9] V. Taranalli, H. Uchikawa, and P. H. Siegel, "Error analysis and intercell interference mitigation in multi-level cell flash memories," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 1868–1873.

[10] 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.

[11] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[12] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge: Cambridge University Press, 2004.

[13] K. A. S. Immink, "Runlength-limited sequences," *Proc. IEEE*, vol. 78, pp. 1745–1759, Nov. 1990.

[14] A. Kato and K. Zeger, "On the capacity of two-dimensional run-length constrained channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1527–1540, Jul. 1999.

[15] B. Marcus, R. Roth, and P. Siegel, *Constrained systems and coding for recording channels*. Technion-I.I.T., Department of Computer Science, 1998.

[16] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.

[17] 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.

[18] A. Bassalygo, "Correcting codes with an additional property," *Probl. Inf. Transm.*, vol. 4, no. 1, pp. 1–5, 1968.

[19] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Tech.*, vol. 23, p. 46334634, Mar. 1981.

[20] M. Mansuripur, "Enumerative modulation coding with arbitrary constraints and postmodulation error correction coding for data storage systems," in *Proc. SPIE*, 1991, p. 1499.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] S. W. Golomb, "Undecidability and nonperiodicity for tilings of the plane," *Inventiones Math.*, vol. 12, pp. 177–209, 1971.

[26] R. Berger, *The undecidability of the Domino problem*. American Mathematical Society, 1966.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] J. Ashley and B. Marcus, "Two-dimensional low-pass filtering codes," *IEEE Trans Commun.*, vol. 46, no. 6, pp. 724–727, 1998.

[41] 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.

[42] H. Pfister, J. B. Soriaga, and P. Siegel, "On the achievable information rates of finite state ISI channels," in *IEEE Glob. Telecommun. Conf.*, vol. 5, 2001, pp. 2992–2996.

[43] 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.

[44] 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.

[45] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics-Doklady*, vol. 10, no. 8, Feb. 1966.

[46] 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.

[47] P. Pakzad and V. Anantharam, "Estimation and marginalization using Kikuchi approximation methods," *Neural Computation*, vol. 17, pp. 1836–1873, 2003.

[48] V. Chvatal and B. Reed, "Mick gets some the odds are on his side," *Proc. of 33rd FOCS*, 1992.

[49] O. Dubois, Y. Boufkhad, and J. Mandler, "Typical random 3-SAT formulae and the satisfiability threshold," *Proc. of 11th SODA*, pp. 126–127, 2000.

[50] E. Friedgut, "Neccesary and sufficient conditions for sharp threhsolds of graph properties and the k-problem," *Journal of American Math Society*, vol. 12, pp. 1017–1054, 1999.

[51] A. Goerdt, "A remark on random 2-SAT," *Journal Computer System and Sciences*, vol. 53, pp. 469–486, 1996.

[52] 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.

[53] S. Kullback, *Information Theory and Statistics*. New York, NY,: Dover Publications, 1968.

[54] F. Haddadpour, M. J. Siavoshani, and M. Noshad, "Low-complexity stochastic generalized belief propagation," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2016, pp. 785–789.