Enhancing Binary Images of Non-Binary LDPC Codes

Aman Bhatia, Aravind R. Iyengar, and Paul H. Siegel
University of California, San Diego, La Jolla, CA 92093 – 0401, USA
Email: {a1bhatia, aravind, psiegel}@ucsd.edu

Abstract—We investigate the reasons behind the superior performance of belief propagation decoding of non-binary LDPC codes over their binary images when the transmission occurs over the binary erasure channel. We show that although decoding over the binary image has lower complexity, it has worse performance owing to its larger number of stopping sets relative to the original non-binary code. We propose a method to find redundant parity-checks of the binary image that eliminate these additional stopping sets, so that we achieve performance comparable to that of the original non-binary LDPC code with lower decoding complexity.

I. INTRODUCTION

Low-density parity-check (LDPC) codes were introduced by Gallager [1] and rediscovered in the 1990’s. Davey and MacKay showed that non-binary LDPC (NB-LDPC) codes perform better than binary codes for the same bit length [2]. Despite their better performance, higher decoding complexity has limited the implementation of non-binary LDPC codes in real-world applications. Reduced complexity decoding is an ongoing area of research due to its limited implementation in real-world applications. Reduced complexity decoding is an ongoing area of research due to its limited implementation in real-world applications.

We summarize our findings in Section VI.

II. GF($2^b$)-LDPC CODES

An LDPC code $C_q$ over $GF(q)$ where $q = 2^b$ is specified by a parity-check matrix $H_q$ of dimension $m_q \times n_q$ whose elements are from $GF(q)$, where the number of non-zero elements in $H_q$ is proportional to $n_q$. The code is then defined as the nullspace of $H_q$, i.e.,

$$C_q = \{ X \in GF(q)^{n_q} : H_q \otimes_q X^T = 0^T \}$$

where $X$ is assumed to be a row-vector, $0$ the all-zero row-vector, $\otimes_q$ denotes matrix multiplication when matrix elements belong to $GF(q)$, and $^T$ denotes transposition. When the parity-check matrix $H_q$ is full-rank, the code $C_q$ is a vector-subspace of $GF(q)^{n_q}$ of dimension $k_q = n_q - m_q$, has a block-length $n_q$ symbols, and is therefore of rate $R = \frac{k_q}{n_q} = 1 - \frac{m_q}{n_q}$.

We associate with the matrix $H_q$ a bipartite graph, called the Tanner graph $G_q$, as follows. Corresponding to the $j$th column of $H_q$ is a variable node $V_j^q$, and corresponding to the $j$th row is a check node $C_q^i$, $i \in [m_q] \triangleq \{1, 2, \ldots, m_q\}, j \in [n_q]$. Every non-zero entry $h_{i,j}$ of $H_q$ corresponds to an edge between $C_q^i$ and $V_j^q$ with label $h_{i,j}$. We denote by $N^v(j)$ the set of check nodes connected to a given variable node $j \in [n_q]$, i.e., the neighbors of the $j$th variable node, and by $N^c(i)$ the set of variable nodes connected to a given check node $i \in [m_q]$. When the degree of every variable node and every check node in $G_q$ is $d_v$ and $d_c$, respectively, i.e., $|N^v(j)| = d_v$, $|N^c(i)| = d_c$, $\forall \ i \in [m_q], j \in [n_q]$, the LDPC code $C_q$ is said to be $(d_v, d_c)$-regular. It is easy to see that for a $(d_v, d_c)$-regular code, the rate satisfies $R = 1 - \frac{m_q}{n_q} = 1 - \frac{d_v}{d_c}$.

Belief Propagation Decoding

The messages passed around on the Tanner graphs in the BP decoder represent the a posteriori probabilities of the symbols of $GF(q)$. We will briefly describe the BP decoding of non-binary LDPC codes over the binary erasure channel (BEC) using the analogue of the peeling decoder, which for the BEC is the same as the BP decoder over $G_q$, which we denote as BP($H_q$).

We assume a fixed isomorphism $\Phi_B : GF(q) \rightarrow GF(2)^b$ that preserves the addition operation defined on $GF(q)$. This isomorphism is used to map codewords $X \in C_q \subset GF(q)^{n_q}$ binary vectors $x \in GF(2)^{b n_q}$. The vector $x$ is transmitted over a BEC with erasure probability $\epsilon$ and let $E$ be the index set of the erasures in the received word. At the receiver, the a priori set of eligible symbols $Q^{(0)}_j$ for the $j$th variable node consists of symbols which fit the received binary sequence corresponding to the $j$th transmitted symbol. The peeling decoder updates these sets iteratively by exchanging messages between variable and check nodes. For the BEC, the messages passed around iteratively can be represented by sets of eligible symbols. Let $Q_{i,j}^{(\ell)}$ denote the set of eligible symbols sent by the $i$th check node to the $j$th variable node in the $\ell$th iteration. Then, the peeling decoder performs the following steps iteratively for $\ell \geq 1$.
Check node processing: For check node $C^q_i$, $i \in [m_q],$
\[ Q_{i,j}^{(t)} = \sum_{j' \in N^c(i) \setminus \{j\}} h_{i,j'} Q_{j'}^{(t-1)} \quad \forall j \in N^c(i). \]

Variable node processing: For variable node $V^q_j$, $j \in [n_q],$
\[ Q_j^{(t)} = Q_j^{(t-1)} \cap \left( \bigcap_{i \in N^t(j)} h_{i,j}^{-1} Q_i^{(t)} \right) \]
where, if $h \in GF(q)$ and $Q_1, Q_2 \subseteq GF(q)$, we define
\[ hQ \triangleq \{ ha \mid a \in Q \} \]
\[ Q_1 + Q_2 \triangleq \{ a_1 + a_2 \mid a_1 \in Q_1, a_2 \in Q_2 \}. \]

The decoder stops when $|Q_j^{(t)}| = |Q_j^{(t-1)}| \forall j \in [n_q]$. Decoding is successful if $|Q_j^{(t)}| = 1 \forall j \in [n_q]$ for some $\ell \in \mathbb{N}$. It should be noted that any set of eligible symbols ($Q_j^{(t)}$ or $Q_i^{(t)}$) is a coset of a vector subspace of $GF(q)$.

III. Binary Images

Since codes defined over $GF(2^b)$ can be written as a collection of vectors over $GF(2)$, we can consider the binary images of these codes. Further, since BP decoding for $GF(q)$ has a high computational complexity, we can make use of the binary images to decode the non-binary code.

A. Basic Binary Image

The isomorphism $\Phi_B$ defined in Section III maps any row-vector of symbols $X = (X_1, \ldots, X_{n_q}) \in GF(q)^{n_q}$ to a binary row-vector $\Phi_B(X) = (\Phi_B(X_1), \ldots, \Phi_B(X_{n_q}))$ of length $m_q$. The basic binary image of $C_q$ is defined as the set $C_B \subseteq GF(2)^{m_q}$ where $C_B = \{ \Phi_B(X) : X \in C_q \}$. As described earlier, the codewords of $C_B$ are the ones that are transmitted over the BEC. Note that there are $2^{b-1} (2^{b}-2)$ possible choices for the isomorphism $\Phi_B$.

Once fixed, the isomorphism $\Phi_B$ identifies a mapping $\Psi_m : GF(q) \mapsto M_B$, where $M_B$ is the collection of all invertible matrices over $GF(2)$ of size $b \times b$, defined below. For any $h \in GF(q)$,
\[ \Psi_m(h) = \begin{pmatrix} \Phi_B(h \times \Phi_B^{-1}(u_{1,b})) \\ \Phi_B(h \times \Phi_B^{-1}(u_{2,b})) \\ \vdots \\ \Phi_B(h \times \Phi_B^{-1}(u_{b,b})) \end{pmatrix}^T \]
where $u_{i,b}$ denotes the binary row-vector of length $b$ with the element $i$ as 1 and other elements equal to 0, i.e., the unit vector along the $i^{th}$ dimension. The set $\{ \Psi_m(h) : h \in GF(q) \} \subseteq M_B$ constitutes a field under matrix addition and matrix multiplication operations over $GF(2)$, i.e., for $h_1, h_2 \in GF(q)$, $\Psi_m(h_1 + h_2) = \Psi_m(h_1) \oplus \Psi_m(h_2)$ and $\Psi_m(h_1 h_2) = \Psi_m(h_1) \otimes \Psi_m(h_2)$. Also, it is easy to see that for any $h, x \in GF(q)$, $\Psi_m(h) \otimes \Phi_B(x)^T = \Phi_B(hx)^T$.

The mappings $\Phi_B$ and $\Psi_m$ allow us to identify $C_B$ as a linear block code with parity-check matrix $H_B$ that is obtained by replacing each element of $H_q$ by its image given by $\Psi_m$. We could perform BP on the Tanner graph of $H_B$, denoted by BP($H_B$), instead of the more complex BP($H_q$). However this has worse performance in general due to the large number of short cycles introduced by $\Psi_m(h)$’s in the parity-check matrix $H_B$. In particular, for the BEC, we shall show later that this bad performance can be attributed to a larger number of stopping sets of $H_B$.

B. Extended Binary Image

The observation made at the end of the previous subsection leads us to the question whether it is possible to design a binary image of a non-binary code that matches the performance of the non-binary code. This question was answered in the affirmative by Savin in [3] for the BEC. We briefly describe the construction of this code, called the extended binary image.

We define a mapping $\Phi_E : GF(q) \mapsto GF(2)^{2q-1}$ such that each symbol of $GF(q)$ is mapped to a codeword of the simplex code $C_S(b)$ of length $2^b - 1$, i.e., the dual of the Hamming code of blocklength $2^b - 1$. For $X \in GF(q)$,
\[ \Phi_E(X) \triangleq \Phi_B(X) \otimes_2 G_S(b) \]
where $G_S(b)$ is the generator matrix of the simplex code of size $b \times (2^b - 1)$. By linearity of $\Phi_B$, $\Phi_E(X_1 + X_2) = \Phi_E(X_1) \oplus_2 \Phi_E(X_2)$. Note that the columns of $G_S(b)$ are all vectors of weight $1 \leq w \leq b$. We will let the ith column of $G_S(b)$ be $(g_{1,i}, g_{2,i}, \ldots, g_{b,i})^T$ where $i = \sum_{j=1}^{b} g_{j,i} 2^{j-1}$ for $i \in [2^b - 1]$. We use the mapping $\Phi_E$ to map row-vector of symbols $X = (X_1, \ldots, X_{n_q}) \in GF(q)^{n_q}$ to a binary row-vector $\Phi_E(X) \triangleq (\Phi_E(X_1), \ldots, \Phi_E(X_{n_q}))$ of length $(2^b - 1)n_q$. The ordering of the columns of $G_S(b)$ implies that for $i \in [n_q], j \in [b]$
\[ (\Phi_E(X_i))_j = (\Phi_E(X_i))_{2j-1}, \]
where we write $(\Phi_E(X_i))_e$ to denote the e"ith element of the vector representing $\Phi_E(X_i)$ for $X_i \in GF(q)$. The extended binary image of $C_q$ is defined as the set $C_E \subseteq GF(2)^{(2^b-1)n_q}$ where $C_E = \{ \Phi_E(X) : X \in C_q \}$. For a fixed $\Phi_E$, one can define a mapping $\Psi_M : GF(q) \mapsto M_{2b-1}$ such that
\[ \Psi_M(h) \otimes_2 \Phi_E(X)^T = \Phi_E(hX)^T \quad \forall h, X \in GF(q). \]

Then the following can be shown.

Lemma 1: $\Psi_M(h)$ is a permutation matrix for all $h \in GF(q)$.

The matrix $H_{PM}$ of dimension $m_q (2^b - 1) \times n_q (2^b - 1)$ is defined as one obtained by replacing each element of $H_q$ by its image under $\Psi_M$. Then, the Tanner graph of $H_{PM}$ is a graph cover [5] of $G_q$ with all edge labels equal to 1. It is easy to show that the extended binary image is
\[ C_E = \{ x \in GF(2)^{(2^b-1)n_q} : H_{PM} \otimes_2 x^T = 0^T \} \cap C_S(b)^{n_q} \]
where
\[ C_S(b)^{n_q} = C_S(b) \times \ldots \times C_S(b) \subset GF(2)^{(2^b-1)n_q}. \]
The overall parity-check matrix for the extended binary code is therefore of the form

\[ \mathbf{H}_E = \begin{pmatrix} \mathbf{H}_{PM} & \mathbf{I}_{n_q} \odot \mathbf{H}_S(b) \end{pmatrix} \]

where \( \mathbf{I}_{n_q} \) is the \((n_q \times n_q)\) identity matrix, \( \mathbf{H}_S(b) \) is the parity-check matrix of the simplex code and \( \odot \) represents the Kronecker product.

As a consequence of Equation (1), transmitting the code-words of \( \mathcal{C}_B \) amounts to transmitting bits indexed by the set \( \mathcal{I}_t = \{ i \in \{ (2^b - 1) n_q \} : i = 2^j \mod (2^b - 1), j \in \{0, 1, \ldots, b - 1\} \} \) of the corresponding codewords in \( \mathcal{C}_E \) and puncturing the rest. We will let \( \mathcal{I}_p = [(2^b - 1) n_q] \setminus \mathcal{I}_t \) denote the index set of punctured bits in \( \mathcal{C}_E \). For ease of notation, we will define \( x^{B}_{i,j} \equiv \Phi_B(X_i) \) and \( x^{E}_{i,j} \equiv \Phi_E(X_i) \).

**Example 2:** Let us consider a code over \( \text{GF}(8) \) given by the following parity-check matrix \( \mathbf{H}_B = (\alpha \quad \alpha^2 \quad 1) \), where \( \alpha \) is a primitive element of \( \text{GF}(8) \) with the primitive polynomial being \( x^3 + x + 1 = 0 \). The code may be represented as solutions to the following linear constraint \( \alpha X_1 + \alpha^2 X_2 + X_3 = 0 \). Table I gives the chosen mapping \( \Phi_B \). Then the parity equations for the basic binary code can be written as

\[
\begin{bmatrix}
0 & 1 & 1 \\
1 & 1 & 1 \\
1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x^{B}_{1,1} \\
x^{B}_{1,2} \\
x^{B}_{1,3}
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x^{B}_{2,1} \\
x^{B}_{2,2} \\
x^{B}_{2,3}
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}, \text{i.e.,}
\]

\[
\begin{align*}
(x^{B}_{1,2} + x^{B}_{1,3}) + x^{B}_{2,2} + x^{B}_{3,1} &= 0, \\
(x^{B}_{1,1} + x^{B}_{1,2} + x^{B}_{1,3}) + x^{B}_{2,3} + x^{B}_{3,2} &= 0, \\
(x^{B}_{1,1} + x^{B}_{1,3}) + (x^{B}_{2,1} + x^{B}_{2,2}) + x^{B}_{3,3} &= 0.
\end{align*}
\]

The parity-check equation for the extended binary image can be written as

\[
\Psi_M(\alpha) \Phi_E(X_1) + \Psi_M(\alpha^2) \Phi_E(X_2) + \Psi_M(1) \Phi_E(X_3) = 0.
\]

Thus, the seven binary parity-check equations represented by the equation above are represented by \( \mathbf{H}_{PM} \) given by

| \( j \) | \( \Phi_E(X_1)_j \) | \( \Phi_E(X_2)_j \) | \( \Phi_E(X_3)_j \) |
|---|---|---|---|
| \( P_1 \) | 0 | 1 | 1 |
| \( P_2 \) | 0 | 0 | 1 |
| \( P_3 \) | 1 | 0 | 0 |
| \( P_4 \) | 0 | 1 | 0 |
| \( P_5 \) | 0 | 0 | 1 |
| \( P_6 \) | 0 | 0 | 1 |
| \( P_7 \) | 0 | 0 | 1 |

Here, equation \( P_1 \) is \( x^{E}_{1,6} + x^{E}_{2,2} + x^{E}_{3,1} = 0 \) which together with the simplex constraints \( x^{E}_{1,6} = x^{B}_{1,2} + x^{B}_{1,3}, x^{E}_{2,2} = x^{B}_{2,2} \) and \( x^{E}_{3,1} = x^{B}_{3,1} \) can be written as Equation (2). Similarly, equations \( P_3 \) and \( P_4 \) correspond to Equations (3) and (4) respectively. Equations \( P_5, P_6, P_7 \) are all linear combinations of the above equations.

In [3], it was shown that the extended binary image was equivalent to the non-binary code from which it was constructed. But this equivalence was established for the BEC with BP updates at the parity-check nodes and ML updates at the simplex nodes. In order to achieve this equivalence with BP, we will assume that the parity-check matrices representing the simplex codes have redundant parity-checks assuring ML performance with BP. For the simplex codes, this is achieved by forming the parity-check matrix with rows of weight 3 [6]. Hence, \( \mathbf{H}_S(b) \) is the parity-check matrix of size \((2^b - 1)(2^{b-1} - 1) \times (2^b - 1)\) for the simplex code. We will denote BP over the Tanner graph of \( \mathbf{H}_E \) with \( \mathbf{H}_S(b) \) having redundant checks and when only bits indexed by \( \mathcal{I}_t \) are transmitted as \( \text{BP}(\mathbf{H}_E) \).

**IV. SUPERIORITY OF BP(\( \mathbf{H}_q \)) OVER BP(\( \mathbf{H}_B \))**

From the construction of the basic binary image and the extended binary image, it is clear that for each parity-check equation in \( \mathbf{H}_q \), there are \( b \) parity-check equations in \( \mathbf{H}_B \) and \((2^b - 1)\) parity-check equations in the \( \mathbf{H}_{PM} \) portion of \( \mathbf{H}_E \).

**Lemma 3:** For each row \( \mathbf{h}_q \) in the parity-check matrix \( \mathbf{H}_q \), the parity-check matrix \( \mathbf{H}_E \) contains all non-trivial linear combinations of the \( b \) rows corresponding to \( \mathbf{h}_q \) in the parity-check matrix \( \mathbf{H}_B \).

**Discussion:** First, note that the extra variables in the extended binary image were defined such that all non-trivial linear combinations of the transmitted bits were represented by a variable and these linear relations were maintained through the use of the simplex code for each non-binary symbol. Second, observe that by using the permutation matrices, we obtained a set of \((2^b - 1)\) parity-check equations each involving exactly one of the variables from each simplex codeword. Putting these two together, we can show the above result.

Since it is known that the performance of \( \text{BP}(\mathbf{H}_q) \) is the same as that of \( \text{BP}(\mathbf{H}_E) \), it suffices to show that \( \text{BP}(\mathbf{H}_E) \) has superior performance in comparison with \( \text{BP}(\mathbf{H}_B) \). The performance of codes over the BEC with BP decoding is determined by the stopping sets of the Tanner graph of their parity-check matrix [7]. By an erasure pattern \( \mathcal{E} \), we mean the index set of the erasures in a vector. For a binary parity-check
matrix $H$, let $\beta(H, E)$ denote the erasure pattern obtained at the end of decoding the received word with erasure pattern $E$ using BP on the Tanner graph of $H$. Then a stopping set is an erasure pattern $E$ such that $\beta(H, E) = E$. Let $S(E) = \{ E : \beta(H, E) = E \}$ be the set of stopping sets of $H$. Notice that this condition is the same as the graph-theoretic requirement that every check node neighbor of variables in the Tanner graph of $H$ indexed by $E$ be connected to these variables at least twice. However, with the extended binary image, we are interested in erasure patterns only among the transmitted variables. Hence we do not have a corresponding graph-theoretic definition of stopping sets of $BP(H_E)$.

**Definition 4 (Stopping Sets of $BP(H_E)$):** Consider $BP$ decoding of the extended binary code over the BEC with the parity-check matrix $H_E$. A stopping set, $E$, is a subset of the index set of transmitted bits such that $BP$ decoding can recover no transmitted bit in $E$, i.e.,

$$E \subseteq I_t \text{ such that } \beta(H_E, E \cup I_p) \cap I_t = E.$$

**Discussion:** Let us denote by $S_E$ the set of all stopping sets of the extended binary image as defined above. Let $S'(H_E)$ denote the set of all stopping sets of the code with parity-check matrix $H_E$ assuming all bits are transmitted. Define

$$S'(H_E) = \{ S \cap I_t : S \in S(H_E) \}.$$

Then, $S_E = S'(H_E)$. This is because for each $S \in S'(H_E)$, $\beta(H_E, S)$ is the maximal stopping set $S'$ in $S(H_E)$ such that $S' \cap I_t = S$. Note that for the basic binary image, the set of all stopping sets is just the normal definition $S_B = S(H_B)$ since every bit of the basic binary image is transmitted.

**Proposition 5 (BP($H_E$) better than BP($H_B$)):** $S_E \subseteq S_B$.

**Proof:** Suppose to the contrary that $\exists S \in S_E \setminus S_B$. Then, there is a parity-check equation in $H_B$ that involves only one erasure, say $x_{i,e}$. Let this equation be

$$\sum_{j \in I_t} (\Phi_B(X_i))_j + \sum_{k \in I_p} \sum_{j \in I_k} (\Phi_B(X_k))_j = 0 \implies \sum_{j \in I_t} x_{i,j}^E + \sum_{j \in I_k} x_{k,j}^E = 0 \quad (5)$$

for some index sets $I_t, I_k \subseteq \{0, 1\}$, $\forall k \in I_k, I \subseteq \{n\} \setminus \{i\}$ and $\epsilon \in I_t$. We know from the construction of the extended binary image that there exist variables

$$x_{k, \Sigma}^E = (\Phi(E(X_k)))_{\sum_{j \in I_k} 2^{j-1}} = \sum_{j \in I_k} (\Phi(B(X_k)))_j = \sum_{j \in I_k} (\Phi(E(X_k)))_{2^{j-1}, k \in I \cup \{i\}}.$$

From Lemma 3 we also know that the equation

$$x_{i, \Sigma}^E + \sum_{k \in I_p} x_{k, \Sigma}^E = 0 \quad (6)$$

is contained in $H_{PM}$. Since every bit except $x_{i,e}$ was known in Equation (5), every bit except $x_{i,\Sigma}$ is known in Equation (6). Hence the extended binary image can solve for $x_{i,\Sigma}^E$. But the simplex portion of $H_E$ contains the equation

$$x_{i,\Sigma}^E = \sum_{j \in I_t} (\Phi_B(X_i))_j \sum_{j \in I_k} (\Phi_B(X_k))_j$$

which has only one unknown $x_{i,\Sigma}^E = x_{i,e}^B$, which can also be solved for. This is a contradiction to the assumption that $S \in S_E$. Therefore, $S_E \setminus S_B = \emptyset$.

**Example 6:** Consider the code in Example 2 and let the received word be $(?0? 0?0 000)$, where $?$ denotes an erasure. It is easy to see that this received word corresponds to a stopping set of the basic binary image. However, the extended binary image can recover all erasures. Using $P_7$ we can recover $x_{2,7}^E$, then using the simplex equation $x_{2,7}^E = x_{2,1}^E + x_{2,2}^E + x_{2,4}^E$ we obtain $x_{2,1}^E$. In turn, $x_{2,1}^E, \forall i \in \{1, \ldots, 7\}$ can be obtained. Finally we can use $P_3$ and $P_7$ to recover $x_{1,1}^E$ and $x_{1,4}^E$.

Note that although we have only shown improper containment of $S_E$ in the set of stopping sets of $H_B$, in most cases this containment is proper, i.e., $S_E \subset S_B$. One case where $S_E = S_B$ is when the Tanner graph corresponding to $H_B$ is cycle-free.

**Proposition 5** gives us an analytical insight into why non-binary codes perform well. However, since the superiority is established only in comparison with the basic binary image corresponding to the non-binary code, the result constitutes only a partial answer to why non-binary codes are better than their binary counterparts in general.

V. ENHANCING $C_B$ USING REDUNDANT PARITY-CHECKS

It is known that the performance of a linear code over the BEC with BP can be improved by using a parity-check matrix with redundant parity-checks (RPCs). In fact, by adding enough parity-checks, we can guarantee ML performance with BP (See [6] and references therein). A similar notion is that of stopping redundancy [8], where RPCs are added to remove stopping sets of size smaller than the minimum distance of the code. In the same spirit, we pose the question whether it is possible to achieve the performance of a non-binary code by its basic binary image with some RPCs.

**Proposition 7:** For every $S \in S_B \setminus S_E$, there exists an RPC for the basic binary image, $h_S \otimes \Phi_B(X)^T = 0$, such that $S \notin S(H_B)$ where $H_B^T = (H_B^T | h_S^T)$.

**Proof:** Since $S \notin S_E$, the BP decoder working over $H_E$ can solve for some erased transmitted bit $x_{i,\Sigma}^E = x_{i,e}^B$. This implies that the ML decoder working with $H_B$ can also solve for $x_{i,e}^B$. Since ML decoding over the BEC is the same as Gaussian elimination, the above means that there exists a linear combination of the parity-check equations in $H_B$ which has only $x_{i,e}$ as the unknown, which can be set as $h_S$.

Note that since there might be multiple linear combinations of parity-check equations of $H_B$ that can solve for $x_{i,e}$, the RPC $h_S$ is not always unique.

We now describe an algorithm that finds the RPC $h_S$, pseudocode for which is given as Algorithm 1. Given $S$, the algorithm starts the peeling decoding over $H_E$ with the erasure pattern $S \cup I_p$. The decoder maintains a list of all the punctured
Algorithm 1 Algorithm to find \( h_S \) given \( S \in S_B \setminus S_E \)

1: \( R \leftarrow \emptyset, \mathcal{L} \leftarrow \emptyset, h_S \leftarrow 0 \)
2: while \( \exists \) a row \( h_{i,j}^E \in H_E \) that solves \((\Phi_E(X))_i\) do
3: \( R \leftarrow R \cup \{(i,j)\} \)
4: if \( i \in I_p \) then \( \mathcal{L} \leftarrow \{(i,j)\} \)
5: while \( \mathcal{L} \neq \emptyset \) do
6: \( (i,j) \leftarrow \text{pop}(\mathcal{L}) \)
7: Let \( I_j \subset \{(2^b - 1)n_q : \sum k \in I_j \cup \nu_k, (2^b - 1)n_q = h_{i,j}^E \} \)
8: for all \( i' \in I_j \cap I_p \setminus \{i\} \) do
9: \( \text{Locate } (i', j') \in R \)
10: \( \text{push}(\mathcal{L}, (i', j')) \)
11: end for
12: end for
13: for all \( k \in I_j \cap I_p \) do
14: \( \sigma_k \leftarrow \lfloor \frac{1}{2^b} \rfloor b + (k \mod (2^b - 1)) \)
15: \( h_S \leftarrow h_S \oplus 2 \cup \nu_{\sigma_k, bn_q} \)
16: end for
17: end while
18: return \( h_S \)
19: terminate
20: end if
21: end while

Algorithm to find \( h_S \) given \( S \in S_B \setminus S_E \)

Algorithm for finding \( h_S \) given \( S \in S_B \setminus S_E \) involves initializing the recovery set \( R \) and the variable set \( \mathcal{L} \) to empty sets, and the parameter \( h_S \) to 0. It iterates through all rows \( h_{i,j}^E \) of the extended binary image that solve the equation \((\Phi_E(X))_i\), and for each such row, it extracts the corresponding variable set \( I_j \), which is a subset of \( (2^b - 1)n_q \) for some \( b \). For each variable \( k \in I_j \cap I_p \), it calculates a parameter \( \sigma_k \) and updates the parameter \( h_S \) accordingly. The process is repeated until all \( h_{i,j}^E \) have been processed, and the final \( h_S \) is returned.

### Tables

#### Table II: Stopping Set Weights for a GF(4) Code, \( C_4 \), and Its Binary Images

| \( w \) | \( A_B^w \) | \( |S_E^w| \) | \( |S^w(H_B)| \) | \( |S^w(H_b^4)| \) | \( |S^w(H_b^8)| \) |
|---|---|---|---|---|---|
| \( \leq 4 \) | 0 | 0 | 0 | 0 | 0 |
| 5 | 4 | 4 | 4 | 4 | 0 |
| 6 | 2 | 2 | 11 | 2 | 2 |
| 7 | 7 | 7 | 31 | 7 | 7 |
| 8 | 7 | 9 | 50 | 9 | 9 |
| 9 | 3 | 4 | 86 | 4 | 4 |
| 10 | 12 | 23 | 171 | 23 | 23 |
| 11 | 22 | 30 | 343 | 30 | 30 |
| 12 | 63 | 80 | 873 | 80 | 80 |
| 13 | 87 | 120 | 2199 | 151 | 120 |
| 14 | 122 | 204 | 5463 | 287 | 204 |
| 15 | 205 | 418 | 13891 | 650 | 418 |
| 16 | 369 | 793 | 35209 | 1322 | 821 |

#### Table III: Stopping Set Weights for a GF(8) Code, \( C_8 \), and Its Binary Images

| \( w \) | \( A_B^w \) | \( |S_E^w| \) | \( |S^w(H_B)| \) | \( |S^w(H_b^8)| \) | \( |S^w(H_b^16)| \) |
|---|---|---|---|---|---|
| \( \leq 2 \) | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 59 | 1 | 1 |
| 7 | 5 | 5 | 82 | 5 | 5 |
| 8 | 13 | 13 | 509 | 13 | 13 |
| 9 | 38 | 38 | 2781 | 156 | 38 |
| 10 | 64 | 64 | 11763 | 612 | 64 |
| 11 | 143 | 147 | 45310 | 2102 | 310 |
| 12 | 309 | 358 | 169120 | 6363 | 871 |
| 13 | 799 | 970 | 663617 | 19346 | 2575 |
| 14 | 1906 | 2525 | 2727519 | 62032 | 7893 |

For the codes under consideration, Figures 1 and 2 plot the BP performance of the non-binary codes, their basic binary...
images, and the enhanced binary images. The improvement in performance with RPCs is evident in the figures. The FERs of the non-binary code and the enhanced binary image are close to each other, and the corresponding BERs are also comparable.

VI. CONCLUSION

We showed that when the transmission occurs over the BEC, BP decoding over the non-binary graph has a better performance than BP decoding over the Tanner graph of the basic binary image of the code. We proposed an algorithm to efficiently find effective redundant parity-checks for the basic binary image by observation of the BP decoding iterations of the extended binary image. Through numerical results and simulations, the effectiveness of the proposed algorithm was established. Obtaining bounds on the number of RPCs of the basic binary image needed to achieve the same performance as BP over non-binary code would be of interest. Similar characterization of the reasons for the superiority of non-binary codes over other channels involving errors as well as erasures will give further insight on designing strong codes and efficient decoding algorithms for such channels.

VII. ACKNOWLEDGMENT

This work was funded in part by NSF Grant CCF-0829865 and by Western Digital. The authors would like to thank Xiaojie Zhang for help with simulations.

REFERENCES

[1] R. G. Gallager, Low Density Parity Check Codes. Cambridge, Massachusetts: MIT Press, 1963.
[2] M. Davey and D. MacKay, “Low density parity check codes over GF(q),” in IEEE Inf. Theory Workshop, Kilnary, Ireland, 22-26 Jun. 1998, pp. 70 – 71.
[3] V. Savin, “Binary linear-time erasure decoding for non-binary LDPC codes,” in 2009 IEEE Information Theory Workshop, Taormina, Italy, 11-16 Oct. 2009.
[4] V. Rathi and R. Urbanke, “Density evolution, thresholds and the stability condition for non-binary LDPC codes,” IEE Proceedings on Communications, vol. 152, no. 6, pp. 1069 – 1074, Dec. 2005.
[5] T. Richardson and R. Urbanke, Modern Coding Theory. Cambridge University Press, 2008.
[6] J. Han and P. Siegel, “On ML redundancy of codes,” in Proc. IEEE Int. Symp. Inf. Theory, Toronto, ON, Canada, 6-11 Jul. 2008, pp. 280 – 284.
[7] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” IEEE Trans. Inf. Theory, vol. 48, no. 6, pp. 1570 – 1579, Jun. 2002.
[8] M. Schwartz and A. Vardy, “On the stopping distance and the stopping redundancy of codes,” IEEE Trans. Inf. Theory, vol. 52, no. 3, pp. 922 – 932, Mar. 2006.
[9] L. P. Sy, V. Savin, and D. Declercq, “Extended non-binary low-density parity-check codes over erasure channels,” CoRR, vol. abs/1103.2691, 2011.
[10] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge-growth Tanner graphs,” IEEE Trans. Inf. Theory, vol. 51, no. 1, pp. 386 – 398, Jan. 2005.
[11] E. Rosnes and O. Ytrehus, “An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices,” IEEE Trans. Inf. Theory, vol. 55, no. 9, pp. 4167 – 4178, Sep. 2009.