Some Results on MAP Decoding of Non-Binary LDPC Codes over the BEC

Vishwambhar Rathi and Iryna Andriyanova

***Abstract*—In this paper, the transmission over the Binary Erasure Channel (BEC) using Non-Binary LDPC (NBLDPC) codes is considered. The concept of peeling decoder and stopping sets is generalized to NBLDPC codes and then used to give a combinatorial characterization of decoding failures of NBLDPC codes, the Belief Propagation (BP) decoder is used. Then, the residual ensemble of codes resulted by the BP decoder is defined and the design rate and the expectation of total number of codewords of the residual ensemble are computed. The decoding failure criterion combined with the density evolution analysis helps us to compute the asymptotic residual degree distribution for NBLDPC codes. Our approach to compute the residual degree distribution on the check node side is not efficient as it is based on enumeration of all the possible connections on the check node side which satisfy the decoding failure criterion. So, the computation of the asymptotic check node side residual degree distribution and further part of our analysis is performed for NBLDPC codes over GF_2^m with $m = 2$. In order to show that asymptotically almost every code in such LDPC ensemble has a rate equal to the design rate, we generalize the argument of the Maxwell construction to NBLDPC codes, defined over GF_2^2. It is also observed that, like in the binary setting, the Maxwell construction, relating the performance of MAP and BP decoding holds in this setting.**

I. Introduction

There are two main research directions in the setting of Non-Binary Low Density Parity Check (NBLDPC) codes. One direction is to analyze performance of NBLDPC codes when transmission takes place over a non-binary input channel [2]–[10]. The other direction, which is of interest to us, is to analyze the performance of NBLDPC codes when transmission takes place over a binary input channel. It was reported in [11] that by having non-binary alphabets, performance of a code under iterative decoding can be improved significantly. After this work, the performance of NBLDPC codes under iterative decoding for transmission over binary input channels has been of interest to many researchers [12]–[16]. Though there is no clear understanding of the iterative decoding performance of NBLDPC codes, there are some partially established results. However, nothing much is known about the performance of NBLDPC codes over the binary input channel under the optimal maximum a posteriori (MAP) decoding. To the best of our knowledge, the only attempt was made in [13], where the density evolution analysis for NBLDPC codes for the BEC and BP decoding was presented. Using the BP estimates, upper bounds on the MAP decoding thresholds were derived which were conjectured to be tight. The main basis of this conjecture was that the Maxwell construction of [17], which relates the MAP and BP performance in the binary setting, should also hold in the non-binary setting.

The aim of this paper is to investigate the MAP decoding performance of NBLDPC codes when transmission takes place over the BEC. Towards this goal, our approach is to generalize the arguments of [17] from the binary to the non-binary setting. In the rest of the paper we will assume that the underlying channel is the binary erasure channel (BEC).

First consider the binary case. The basic idea of [17],

relating the MAP performance to the BP performance is the following. For the binary case there are two decoders, the BP algorithm and the peeling decoder. The peeling decoder was introduced in [18]. Although these two decoders look different, they are in fact identical if their final estimates on bits are considered. The difference between these two decoders is that of the schedule to update erased bits. In the peeling decoder, the erased bits are updated serially. On the other hand, in the BP decoder the erased bits are updated in parallel. The peeling decoder associates to every code and erasure set a residual graph. If the erasure probability is below the BP threshold then the residual graph is empty as all the bits are known almost surely. It was shown in [18] that if the erasure probability is above the BP threshold then for most instances of the erasure set and graph, the residual graph has a degree distribution close to the *average residual degree distribution*. It was also shown that conditioned on the residual degree distribution, the induced probability distribution is uniform over all the graphs with the given degree distribution.

The entropy of the transmitted codeword conditioned on the channel observation is given by the logarithm of the number of codewords which are compatible with the observation. Thus, the idea of bounding the conditional entropy for an erasure probability above the BP threshold is the following (for erasure probability below the BP threshold, the conditional entropy is zero). As we discussed, the peeling decoder almost surely results in an LDPC ensemble, whose degree distribution is given by the average residual degree distribution. Thus the normalized logarithm of the number of codewords which are compatible with the typical channel observations is lower bounded by the design rate of the average residual ensemble. The design rate only gives a lower bound since some check equations might be dependent. A criterion was derived in [17] which, when satisfied, guarantees that the lower bound is tight, i.e., the actual rate of the ensemble is equal to its design rate. We generalize these arguments to the non-binary setting.

Let us consider the non-binary case following the same sequence as in the binary one. Note that the BP decoder for non-binary codes is defined in [13].

We define the peeling decoder and give its decoding failure criterion for the non-binary case. Based on the decoding failure criterion, we define stopping sets in the non-binary setting. We refer to stopping sets as *stopping constellations* to distinguish them from the binary setting. Then we show the equivalence of the peeling and the BP decoder by showing that both of them get stuck in the largest stopping constellation.

We prove that the residual graph resulted by the BP decoder has degree distribution concentrated around the average degree distribution. We also show that conditioned on the degree distribution of the residual graphs, all the graphs which are compatible with this degree distribution have the same probability. Thus, the residual graphs are elements of an appropriately defined residual ensemble whose degree distribution is given by the average residual degree distribution. We show how the average residual degree distribution can be computed by using the fixed points of density evolution of the BP decoder. Our approach to compute the residual degree distribution on the check node side is not efficient as it is based on enumeration of all the possible connections on the check node side which satisfy the decoding failure criterion. So, the computation of the asymptotic check node side residual degree distribution and further part of our analysis is performed for NBLDPC codes over GF_2^m with $m = 2$. We generalize the criterion of [17] to the case of GF_2^2, which, when satisfied shows that almost all the codes in the ensemble have their rate equal

to the design rate. If this criterion is satisfied by the average residual degree distribution then we show that the conditional entropy of the transmitted codeword is equal to the design rate of the average residual ensemble. We also observe that the Maxwell construction, relating the performance of the MAP and the BP decoder, also holds in the setting of NBLDPC codes over GF_2^2.

The paper is organized in the following way. In Section II we give definitions and notations, give an algebraic definition of BP decoder, and define stopping constellation for NBLDPC code. We discuss the peeling decoder, and the equivalence of the peeling and the BP decoder for the NBLDPC codes in Section III. In Section IV we define the non-binary residual ensemble, compute its design rate, and derive the expression for the average total number of codewords in a residual ensemble. We also show that the residual codes resulted by the BP decoder form a residual ensemble whose degree distribution is given by the average residual degree distribution. Section V contains the treatment of the particular case of NBLDPC codes defined over GF_2^2. In this section, we compute the average residual check node degree distribution for NBLDPC ensemble defined over GF_2^2. We also derive a sufficient condition which guarantees that the asymptotic rate of the residual ensemble is equal to its design rate. This allows us to show the equality between the rate of the residual ensemble and the conditional entropy of the transmitted codewords. Finally, we conclude in Section VI with a discussion on possible extensions of our result to the case of GF_2^m with $m > 2$.

II. Preliminaries

We consider transmission over the Binary Erasure Channel (BEC) with erasure probability ε (denoted by BEC(ε)), using a code from the NBLDPC ensemble $\mathrm{EGL}(n,\lambda,\rho,m)$ defined in [13]. In order to define $\mathrm{EGL}(n,\lambda,\rho,m)$, we first define the standard ensemble of bipartite graphs $\mathbb{G}(n,\lambda,\rho)$ in the following definition [19].

Definition 1 (Ensemble of bipartite graphs $\mathbb{G}(n,\lambda,\rho)$): Let λ (ρ) be the degree distribution of the variable (check) nodes from edge perspective. The variable nodes represent the codeword's symbols and the check nodes represent parity check constraints. We denote the blocklength by n which is equal to the number of variable nodes. $\lambda_i(\rho_j)$ denotes the fraction of edges which are connected to a variable (check) node of degree i (j). $\mathbb{G}(n,\lambda,\rho)$ contains all the bipartite graph with degree distribution (λ,ρ) and block length n.

Instead of defining the ensemble $\mathbb{G}(n,\lambda,\rho)$ in terms of degree distribution from edge perspective, we can equivalently define it in terms of degree distribution (Λ,Γ) from node perspective. Λ_i (Γ_j) denotes the fraction of variable (check) nodes with degree i (j).

We denote the set of neighbors of a node x by $\mathcal{N}(x)$. The set of all the neighbors of a node x excluding the neighbor y is denoted by $\mathcal{N}(x)\backslash y$. Before defining $\mathrm{EGL}(n,\lambda,\rho,m)$, we recall that the general linear group GL_2^m over the binary field is the set of all $m\times m$ invertible matrices over the binary field. In other words, GL_2^m is the set of all linear bijective mappings $f:\mathrm{GF}_2^m \mapsto \mathrm{GF}_2^m$.

Definition 2 (Non-binary LDPC ensemble $EGL(n,\lambda,\rho,m)$): For a bipartite graph $\mathtt{G}\in\mathbb{G}(n,\lambda,\rho)$, we label each edge of $\mathtt{G}$ with a linear bijective mapping $f:\mathrm{GF}_2^m \mapsto \mathrm{GF}_2^m$. For a particular edge, f is chosen uniformly at random from GL_2^m and this choice is independent of the choice of mappings for other edges. The variable nodes take values in GF_2^m and a check node $\mathtt{c}$ represent parity

check equations of the form

$$\sum_{i \in \mathcal{N}(\mathtt{c})} f_{i\mathtt{c}}(x_i) = 0,$$

where $f_{i\mathtt{c}}$ is the edge label connecting variable node i and check node $\mathtt{c}$. The NBLDPC ensemble $\mathrm{EGL}(n,\lambda,\rho,m)$ is the set of all such codes defined for all $\forall \mathtt{G} \in \mathbb{G}(n,\lambda,\rho)$.

For degree distribution (Λ,Γ) from node perspective, we denote the equivalent NBLDPC ensemble by $\mathrm{EGL}(n,\Lambda,\Gamma,m)$. In the asymptotic limit of blocklength n, $\mathrm{EGL}(n,\lambda,\rho,m)$ and $\mathrm{EGL}(n,\Lambda,\Gamma,m)$ are denoted by $\mathrm{EGL}(\lambda,\rho,m)$ and $\mathrm{EGL}(\Lambda,\Gamma,m)$.

The design rate of an ensemble is the rate under the assumption that all the parity check constraints are independent. The design rate of $\mathrm{EGL}(n,\Lambda,\Gamma,m)$ is given by

$$R_{\mathrm{des}} = 1 - \frac{\sum_i i\Lambda_i}{\sum_j j\Gamma_j}.$$

We denote a codeword by X and its channel output by Y. The entropy of a codeword conditioned on the channel output is given by $H(X|Y)$. When we want to emphasize the code G to which X belong to, we write $H_G(X|Y)$.

In order to transmit a codeword symbol $x_i \in \mathrm{GF}_2^m$ corresponding to variable node i, we transmit the m binary bits representing the symbol x_i. As we assume transmission over the BEC, the received symbols, which are vectors of length m, contain either known values of bits (0 or 1) or erasures. Hence, the messages of the BP decoder have a very specific form. The messages are real vectors of length 2^m and the α^{th} component of the message gives the a posteriori probability that the corresponding symbol is α, $\alpha \in \mathrm{GF}_2^m$. It was shown in [13] that the error probability performance of the BP decoder is independent of the transmitted codeword. Thus in the rest of the paper, we assume that the all-zero codeword is transmitted unless mentioned otherwise. The messages under the all-zero codeword assumption have the following properties [13]:

1) The non-zero entries of a message are all equal.
2) The indices corresponding to non-zero entries of a message form a subspace of GF_2^m. Thus each message is equivalent to a subspace. This means that if the entries corresponding to α and β, $\alpha,\beta \in \mathrm{GF}_2^m$, are non-zero, then so is the entry corresponding to $\alpha+\beta$.
3) The variable-node side operation is equivalent to taking the intersection of the subspaces corresponding to the incoming messages.
4) The edge operation is equivalent to mapping a subspace of GF_2^m of dimension k to another subspace of GF_2^m of the same dimension.
5) The check-node side operation is equivalent to taking the sum of the subspaces corresponding to the incoming messages.

In general, we note that the messages of the BP decoder are not subspaces of GF_2^m but are its cosets. However, under the all-zero codeword assumption they are indeed subspaces which allows us to simplify the analysis.

Based on the properties listed above, we say that the dimension of a message Ψ, call it $\mathrm{Dim}(\Psi)$, is k if the number of non-zero entries of Ψ is 2^k. By a slight abuse of notation, we denote the subspace of indices corresponding to non-zero entries of the message Ψ also by Ψ. Note that the subspaces corresponding to initial messages have a specific form. An initial message of dimension k has the set of basis vectors $\{e_{i_1},\ldots,e_{i_k}\}$, where e_i is a vector of length m with i^{th} component equal to 1 and the remaining components equal to zero. This corresponds to the message where the bits $i_1,\ldots,i_k$ are erased and the rest of them are known. In such a subspace, the bit $i_j, j \in \{1,\ldots,k\}$ can take both the values 0 and 1 independently of the values of the

remaining $m-1$ variables. We denote the set of such messages by $\mathcal{M}_{\text{ini}}$ and $\mathcal{M}$ denotes the set of all messages of the BP decoder. Note that $\mathcal{M}_{\text{ini}}$ is a strict subset of $\mathcal{M}$.

Using the subspace interpretation of messages for transmission over the BEC, we define the BP decoder in terms of this interpretation. We divide l^{th} iteration of the BP decoder into four stages. We denote a message from a node x to node y in the l^{th} iteration and j^{th} stage by $\Psi_{x,y}^{(l,j)}$, $j \in \{1,2,3,4\}$.

BP Decoder:

Input: Channel output and bipartite graph representation of the NBLDPC code.

If any message in the current iteration is different from its value in the previous iteration, repeat the following four stages.

First Stage: During the l^{th} iteration, where $l > 1$, in the first stage variable node $\mathtt{v}$ computes and sends the message $\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)}$ towards the check node $\mathtt{c}$, $\mathtt{c} \in \mathcal{N}(\mathtt{v})$, which is given by

$$\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)} = C_{\mathtt{v}} \cap \left(\cap_{j \in \mathcal{N}(\mathtt{v}) \setminus \mathtt{c}} \Psi_{j,\mathtt{v}}^{(l-1,4)} \right),$$

where $\Psi_{j,\mathtt{v}}^{(l-1,4)}$ is the message received by $\mathtt{v}$ from the check node j in the 4^{th} (final) stage of previous iteration. If $l = 1$, then $\Psi_{\mathtt{v},\mathtt{c}}^{(1,1)} = C_{\mathtt{v}}$, where $C_{\mathtt{v}}$ is the subspace corresponding to the channel erasures for variable node $\mathtt{v}$.

Second stage: The second stage corresponds to multiplying $\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)}$ by the edge label $f_{\mathtt{vc}}$. We denote the resulting message by $\Psi_{\mathtt{v},\mathtt{c}}^{(l,2)} \triangleq f_{\mathtt{vc}} \Psi_{\mathtt{v},\mathtt{c}}^{(l,1)}$.

Third stage: In the third stage, check node $\mathtt{c}$ computes the message $\Psi_{\mathtt{c},\mathtt{v}}^{(l,3)}$ which is to be sent to the variable node $\mathtt{v}$,

$$\Psi_{\mathtt{c},\mathtt{v}}^{(l,3)} = \sum_{i \in \mathcal{N}(\mathtt{c}) \setminus \mathtt{v}} \Psi_{i,\mathtt{c}}^{(l,2)}.$$

Fourth stage: The fourth and final stage corresponds to multiplying the message $\Psi_{\mathtt{c},\mathtt{v}}^{(l,3)}$ by the inverse of the mapping $f_{\mathtt{vc}}$ and the resulting message is denoted by $\Psi_{\mathtt{c},\mathtt{v}}^{(l,4)} \triangleq f_{\mathtt{vc}}^{-1} \Psi_{\mathtt{c},\mathtt{v}}^{(l,3)}$. The variable node $\mathtt{v}$ is equally likely to take any value in the subspace $B_{\mathtt{v}}^{(l)}$,

$$B_{\mathtt{v}}^{(l)} = C_{\mathtt{v}} \cap \left(\cap_{i \in \mathcal{N}(\mathtt{v})} \Psi_{i,\mathtt{v}}^{(l,4)} \right).$$

Output: Final estimates of the variable nodes $B = \{B_{\mathtt{v}}\}_{\mathtt{v} \in \mathcal{V}}$,

$$B_{\mathtt{v}} = C_{\mathtt{v}} \cap \left(\cap_{i \in \mathcal{N}(\mathtt{v})} \Psi_{i,\mathtt{v}}^{(4)} \right),$$

where we drop the iteration number to indicate that messages have converged. The convergence of the BP decoder is shown in the next section.

Note that at the end of each iteration of the BP decoder, there is a *state assignment* $B^{(l)} = \{B_{\mathtt{v}}^{(l)}\}_{\mathtt{v} \in \mathcal{V}}$ for variable nodes. In general, we define a state assignment $E = \{E_{\mathtt{v}}\}_{\mathtt{v} \in \mathcal{V}}$ to be assignment of a subspace $E_{\mathtt{v}}$ to variable node $\mathtt{v}$ such that $\mathtt{v}$ can only take values belonging to $E_{\mathtt{v}}$. Also, all the variable node values, belonging to $E_{\mathtt{v}}$, have the same probability. We call the output of the BP decoder, $B = \{B_{\mathtt{v}}\}_{\mathtt{v} \in \mathcal{V}}$, the *BP state assignment*. We define the *channel state assignment* C as $C = \{C_{\mathtt{v}}\}_{\mathtt{v} \in \mathcal{V}}$, where $C_{\mathtt{v}}$ is the subspace corresponding to the channel erasures for the variable node $\mathtt{v}$.

Based on the concept of state assignment, we introduce the following terminology which will be useful in analyzing the peeling decoder. We say that a check node $\mathtt{c}$ is *active* with respect to state assignment E if there exists $\mathtt{v} \in \mathcal{N}(\mathtt{c})$ such that $E_{\mathtt{v}} \cap f_{\mathtt{vc}}^{-1} \left(\sum_{i \in \mathcal{N}(\mathtt{c}) \setminus \mathtt{v}} f_{i\mathtt{c}} E_i \right)$ is a strict subset of $E_{\mathtt{v}}$, where $f_{i\mathtt{c}}$ is the mapping on the edge connecting variable node $i \in \mathcal{N}(\mathtt{c})$ to the check node $\mathtt{c}$. We call $\mathtt{v}$ and $\mathtt{c}$ an *active pair*. Also any check node of degree 1 is active if its neighboring variable node has a state of non-zero dimension. It becomes inactive once its connected variable node is assigned the subspace of

dimension zero. We say that a check node $\mathtt{c}$ satisfies the *decoding failure criterion* with respect to the state assignment E if

$$E_{\mathtt{v}} \subseteq f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}}E_i\right), \forall \mathtt{v}\in\mathcal{N}(\mathtt{c}). \qquad (1)$$

In other words, if $\mathtt{c}$ satisfies the decoding failure criterion with respect to the state assignment E then it is inactive with respect to state assignment E. Next, we define the concept of stopping constellation for NBLDPC codes.

Definition 3: A *stopping constellation* is a state assignment $E=\{E_{\mathtt{v}}\}_{\mathtt{v}\in\mathcal{V}}$, where $E_{\mathtt{v}}\in\mathcal{M}_{\text{ini}}$. The state assignment is such that there are no active pairs of nodes, i.e., for every variable node $\mathtt{v}$ and each check node $\mathtt{c}$ connected to $\mathtt{v}$, $E_{\mathtt{v}}\subseteq f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}}E_i\right)$. Thus all the check nodes satisfies decoding failure criterion with respect to E if E is a stopping constellation.

We denote the number of mappings belonging to GL_2^m which map a given subspace of dimension k to another given subspace of dimension k by $g(m,k)$. It is given by:

$$g(m,k)=|\,\mathrm{GL}_2^m\,|\begin{bmatrix} m \\ k \end{bmatrix}^{-1}, \qquad (2)$$

where $\begin{bmatrix} m \\ k \end{bmatrix}$ is the Gaussian binomial coefficient which denotes the number of different subspaces of dimension k of GF_2^m and is given by

$$\begin{bmatrix} m \\ k \end{bmatrix} = \begin{cases} 1, & \text{if} \quad k=0 \ \text{ or } \ k=m, \\ \prod_{l=0}^{k-1}\frac{2^m-2^l}{2^k-2^l}, & \text{otherwise.} \end{cases}$$

We obtain (2) by noticing that the number of elements in GL_2^m which map a subspace V to V_1 is equal to the number of elements in GL_2^m which map V to V_2, where the dimensions of V,V_1 and V_2 are equal to k.

Let A be a subspace of GF_2^m and $f\in\mathrm{GL}_2^m$. Then the subspace fA denotes the image of A under f. In future we will use the two simple facts $f(A\cap B)=(fA)\cap(fB)$, $f(A+B)=(fA)+(fB)$, where $f\in\mathrm{GL}_2^m$ and A, B are subspaces of GF_2^m. Finally, we recall the binomial and multinomial notation

$$\begin{aligned} \binom{n}{j} &\triangleq \frac{n!}{j!(n-j)!}, \\ \binom{n}{j_1,\cdots,j_k} &\triangleq \frac{n!}{(n-\sum_{l=1}^k j_l)!\prod_{l=1}^k j_l!}. \end{aligned}$$

In the next section we define the peeling decoder. Then, we prove that the BP and the peeling decoder get stuck in the largest stopping constellation compatible with the channel output.

III. Peeling Decoder

We define the peeling decoder.

Peeling Decoder:

Input: Channel output and bipartite graph representation of the NBLDPC code.

1) Initialize the state assignment E with the channel state assignment C. Thus, the variable node $\mathtt{v}$ is assigned the state $E_{\mathtt{v}}=C_{\mathtt{v}}$.
2) Take any active pair: let $\mathtt{v}$ be the variable node and $\mathtt{c}$ be the check node. Set $E_{\mathtt{v}}=E_{\mathtt{v}}\cap f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}}E_i\right)$.
3) If there is no active pair, terminate. Otherwise, repeat Step 2.

Output: The final state assignment $E=\{E_{\mathtt{v}}\}_{\mathtt{v}\in\mathcal{V}}$.

Observe that the peeling decoder always terminates as the execution of Step 2 decreases the dimension of the state of $\mathtt{v}$. We show in Lemma 3.1 that the final output does not depend on the choice of active pair in Step 2. Thus, we can choose any active pair. Note that in the binary setting, the possible states of variable nodes are 0 and 1. Hence an active pair of nodes corresponds to a check node which has the corresponding variable

node as the only variable node with state 1 (erased). The remaining attached variable nodes are assigned the state 0 (known). This corresponds to a check node of degree 1 in the setting of the peeling decoder for binary codes [18].

We define the union G of two stopping constellations $E = \{E_{\mathtt{v}}\}_{\mathtt{v}\in\mathcal{V}}$ and $F = \{F_{\mathtt{v}}\}_{\mathtt{v}\in\mathcal{V}}$ as $G_{\mathtt{v}} = E_{\mathtt{v}} + F_{\mathtt{v}}$, $\mathtt{v} \in \mathcal{V}$, where $E_{\mathtt{v}} + F_{\mathtt{v}}$ is the sum of subspaces $E_{\mathtt{v}}, F_{\mathtt{v}}$. We claim that G is also a stopping constellation. We want to prove that $G_{\mathtt{v}} \subseteq f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} G_i\right)$, $\forall \mathtt{c} \in \mathcal{N}(\mathtt{v})$, $\forall \mathtt{v} \in \mathcal{V}$. Now,

$$\begin{aligned} f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} G_i\right) &= f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} (E_i + F_i)\right), \\ &= f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} (f_{i\mathtt{c}} E_i + f_{i\mathtt{c}} F_i)\right), \\ &= f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} E_i\right) \\ &\quad + f_{\mathtt{vc}}^{-1}\left(\sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} F_i\right), \\ &\supseteq E_{\mathtt{v}} + F_{\mathtt{v}} = G_{\mathtt{v}}. \end{aligned}$$

We say that a stopping constellation E is a subset of state assignment S if

$$E_{\mathtt{v}} \subseteq S_{\mathtt{v}}, S_{\mathtt{v}} \in \mathcal{M}_{\text{ini}}, \forall \mathtt{v} \in \mathcal{V}.$$

As the union of two stopping constellations is a stopping constellation, there is a unique largest stopping constellation which is a subset of a state assignment. Note that the largest stopping constellation of a state assignment always exists as the all-zero state assignment is a stopping constellation. Consider the peeling decoder. In what follows we prove that the peeling decoder gets stuck in the largest stopping constellation which is a subset of the channel state assignment.

Lemma 3.1: Consider transmission over the BEC using an NBLDPC code which is decoded by the peeling decoder. After the termination of the peeling decoder, the final state assignment to the variable nodes is the largest stopping constellation E which is a subset of the channel state assignment C.

Proof: Let F be the final state assignment. First note that F is a stopping constellation as this is the only condition for the peeling decoder to terminate. Now $F_{\mathtt{v}} \subseteq C_{\mathtt{v}}$, $\forall \mathtt{v} \in \mathcal{V}$, which follows from the fact that every update of the state of $\mathtt{v}$ by the peeling decoder satisfies this property. We prove by contradiction that F is the largest stopping constellation which is a subset of C. To prove by contradiction, assume that E is the largest stopping constellation. Then F must be a strict subset of E. Otherwise, by taking the union of E and F we could obtain a stopping constellation larger than E, which is also a subset of C. Hence F is a strict subset of E. Now, consider the variable node $\mathtt{v}$ which is the first node whose state changes when acted on by a check node $\mathtt{c}$ such that its intersection with $E_{\mathtt{v}}$ is a strict subset of $E_{\mathtt{v}}$. This is only possible if there are nodes in $\mathcal{N}(\mathtt{c})$ whose state already satisfies this property. But that is not possible as $\mathtt{v}$ is the first node for which this happens. Hence we prove the claim that $F = E$. ■

In the following lemma we show that the BP decoder converges by showing that during an iteration, the message on any edge is a subspace of the message on the same edge during previous iteration.

Lemma 3.2: Consider transmission over the BEC(ε) using NBLDPC code and decoded by the BP decoder. Let $\Psi_{\mathtt{v},\mathtt{c}}^{(l,k)}, k \in \{1,2\}$ (resp. $\Psi_{\mathtt{c},\mathtt{v}}^{(l,k)}, k \in \{3,4\}$) be the message from variable node $\mathtt{v}$ to check node $\mathtt{c}$ (resp. check node $\mathtt{c}$ to variable node $\mathtt{v}$) in the k^{th} stage of the

l^{th} iteration. Then for all edges $\mathtt{vc}$ and $l \geq 2$,

$$\Psi_{\mathtt{v},\mathtt{c}}^{(l,k)} \subseteq \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,k)}, k \in \{1,2\}, \quad \Psi_{\mathtt{c},\mathtt{v}}^{(l,k)} \subseteq \Psi_{\mathtt{c},\mathtt{v}}^{(l-1,k)}, k \in \{3,4\}. \tag{3}$$

Hence the BP decoder converges.

Proof: We first observe that if in the l^{th} iteration, for all the edges $\mathtt{vc}$

$$\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)} \subseteq \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,1)}, \tag{4}$$

then $\Psi_{\mathtt{v},\mathtt{c}}^{(l,2)} \subseteq \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,2)}, \Psi_{\mathtt{c},\mathtt{v}}^{(l,k)} \subseteq \Psi_{\mathtt{c},\mathtt{v}}^{(l-1,k)}, k \in \{3,4\}$. This can be easily seen by noting that

$$\Psi_{\mathtt{v},\mathtt{c}}^{(l,2)} = f_{\mathtt{vc}} \Psi_{\mathtt{v},\mathtt{c}}^{(l,1)} \subseteq f_{\mathtt{vc}} \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,1)} = \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,2)},$$

which implies

$$\Psi_{\mathtt{c},\mathtt{v}}^{(l,3)} = \sum_{i \in \mathcal{N}(\mathtt{c}) \backslash \mathtt{v}} \Psi_{i,\mathtt{c}}^{(l,2)} \subseteq \sum_{i \in \mathcal{N}(\mathtt{c}) \backslash \mathtt{v}} \Psi_{i,\mathtt{c}}^{(l-1,2)} = \Psi_{\mathtt{c},\mathtt{v}}^{(l-1,3)}.$$

Finally, we get for the 4^{th} stage,

$$\Psi_{\mathtt{c},\mathtt{v}}^{(l,4)} = f_{\mathtt{vc}}^{-1} \Psi_{\mathtt{c},\mathtt{v}}^{(l,3)} \subseteq f_{\mathtt{vc}}^{-1} \Psi_{\mathtt{c},\mathtt{v}}^{(l-1,3)} = \Psi_{\mathtt{c},\mathtt{v}}^{(l-1,4)}.$$

Thus we only need to prove that for $l \geq 2$ and all the edges $\mathtt{vc}$, $\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)} \subseteq \Psi_{\mathtt{v},\mathtt{c}}^{(l-1,1)}$. We prove it by induction. First consider the case $l = 2$. Let C be the channel state assignment. Then,

$$\Psi_{\mathtt{v},\mathtt{c}}^{(2,1)} = C_{\mathtt{v}} \cap \left(\cap_{j \in \mathcal{N}_{\mathtt{v}} \backslash \mathtt{c}} \Psi_{j,\mathtt{v}}^{(1,4)} \right) \subseteq C_{\mathtt{v}} = \Psi_{\mathtt{v},\mathtt{c}}^{(1,1)}.$$

Thus our assertion holds true for $l = 2$. Assume that our assertion holds for iteration number l, i.e. (4) is true. Consider the $(l+1)^{\text{th}}$ iteration. By contradiction, assume that there is an edge $\mathtt{v}'\mathtt{c}'$ for which $\Psi_{\mathtt{v}',\mathtt{c}'}^{(l+1,1)} \subsetneq \Psi_{\mathtt{v}',\mathtt{c}'}^{(l,1)}$. This implies that $\exists j \in \mathcal{N}(\mathtt{v}) \backslash \mathtt{c}$ for which $\Psi_{j,\mathtt{v}'}^{(l,4)} \subsetneq \Psi_{j,\mathtt{v}'}^{(l-1,4)}$. This contradicts (4). Hence we prove (3). The convergence of BP now follows trivially. ∎

Before proving that the BP decoder gets stuck in the largest stopping constellation which is a subset of the channel state assignment, we prove the following useful lemma.

Lemma 3.3: Let $E = \{E_{\mathtt{v}}\}$, $\mathtt{v} \in \mathcal{V}$, be the largest stopping constellation contained in the channel state assignment C. Then during BP decoding, the state $E_{\mathtt{v}}$ is a subset of all the outgoing and incoming messages to the variable node $\mathtt{v}$. Also the state assignment E is a subset of the BP state assignment $B = \{B_{\mathtt{v}}\}_{\mathtt{v} \in \mathcal{V}}$.

Proof: Note that the initial message from a variable node $\mathtt{v}$ is $C_{\mathtt{v}}$ and $E_{\mathtt{v}} \subseteq C_{\mathtt{v}}$ by definition of E. Also, the incoming message from a check node $\mathtt{c}$ to a variable node $\mathtt{v}$ in the first iteration is $\Psi_{\mathtt{c},\mathtt{v}}^{(1,4)} = f_{\mathtt{vc}}^{-1} \left(\sum_{i \in \mathcal{N}(\mathtt{c}) \backslash \mathtt{v}} f_{i\mathtt{c}} C_i \right)$. This implies that $\Psi_{\mathtt{c},\mathtt{v}}^{(1,4)}$ contains $E_{\mathtt{v}}$ as $E_i \subseteq C_i$, $i \in \mathcal{N}(\mathtt{c})$ and E forms a stopping constellation. Now, by induction we can prove the desired result. Assume that $\Psi_{\mathtt{v},\mathtt{c}}^{(l-1,1)}$ and $\Psi_{\mathtt{c},\mathtt{v}}^{(l-1,4)}$ contain the state $E_{\mathtt{v}}$, $\forall \mathtt{v} \in \mathcal{V}$, $\forall \mathtt{c} \in \mathcal{N}(\mathtt{v})$. This implies that $\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)}$ and $\Psi_{\mathtt{c},\mathtt{v}}^{(l,4)}$ also contain $E_{\mathtt{v}}$ as

$$\Psi_{\mathtt{v},\mathtt{c}}^{(l,1)} = C_{\mathtt{v}} \cap \left(\cap_{j \in \mathcal{N}(\mathtt{v}) \backslash \mathtt{c}} \Psi_{j,\mathtt{v}}^{(l-1,4)} \right),$$

$$\Psi_{\mathtt{c},\mathtt{v}}^{(l,4)} = f_{\mathtt{vc}}^{-1} \sum_{i \in \mathcal{N}(\mathtt{c}) \backslash \mathtt{v}} f_{i\mathtt{c}} \Psi_{i,\mathtt{c}}^{(l,1)} \supseteq f_{\mathtt{vc}}^{-1} \sum_{i \in \mathcal{N}(\mathtt{c}) \backslash \mathtt{v}} f_{i\mathtt{c}} E_i \supseteq E_{\mathtt{v}},$$

and E is a stopping constellation. The second claim of the lemma can be proved by noting that

$$B_{\mathtt{v}} = C_{\mathtt{v}} \cap \left(\cap_{i \in \mathcal{N}(\mathtt{v})} \Psi_{i,\mathtt{v}}^{(4)} \right)$$

and

$$E_{\mathtt{v}} \subseteq C_{\mathtt{v}}, E_{\mathtt{v}} \subseteq \Psi_{i,\mathtt{v}}^{(4)}, \forall i \in \mathcal{N}(\mathtt{v}).$$

So, we prove that $E_{\mathtt{v}} \subseteq B_{\mathtt{v}}$. Here we have dropped the superscript corresponding to iteration number to denote the converged messages. ∎

From the previous lemma we see that the BP state assignment of a variable node $\mathtt{v}$ satisfies $B_{\mathtt{v}} \supseteq E_{\mathtt{v}}$. If we prove that the BP state assignment also forms a stopping constellation, then $B_{\mathtt{v}} = E_{\mathtt{v}}$, as E is the largest stopping constellation contained in the channel state assignment C, B contains E, and C contains B. In the following

theorem we prove that B is a stopping constellation.

Theorem 3.1: Consider transmission over the BEC using NBLDPC code which is decoded by the BP decoder. Let $\Psi^{(4)}_{\mathtt{c},\mathtt{v}}$ be the incoming message to the variable node $\mathtt{v}$ from check node $\mathtt{c}$ and $\Psi^{(1)}_{\mathtt{v},\mathtt{c}}$ be the message from $\mathtt{v}$ to $\mathtt{c}$ when the BP decoder has converged. Let C be the channel state assignment. Then the BP state assignment $B = \{B_\mathtt{v}\}$, $B_\mathtt{v} = C_\mathtt{v} \cap_{i\in\mathcal{N}(\mathtt{v})} \Psi^{(4)}_{i,\mathtt{v}}, \forall \mathtt{v} \in \mathcal{V}$ is the largest stopping constellation contained in C.

Proof: We need to prove that

$$B_\mathtt{v} \subseteq f^{-1}_{\mathtt{vc}} \sum_{i\in\mathcal{N}(\mathtt{c})\setminus\mathtt{v}} f_{i\mathtt{c}} B_i.$$

For the sake of simplicity assume that the check node $\mathtt{c}$ is of degree 3 and let $\mathcal{N}(\mathtt{c}) = \{\mathtt{v}, \mathtt{v}_1, \mathtt{v}_2\}$. Also note that $B_\mathtt{v} = \Psi^{(4)}_{\mathtt{c},\mathtt{v}} \cap \Psi^{(1)}_{\mathtt{v},\mathtt{c}}$ as $\Psi^{(1)}_{\mathtt{v},\mathtt{c}} = C_\mathtt{v} \cap \left(\cap_{i\in\mathcal{N}(\mathtt{v})\setminus\mathtt{c}} \Psi^{(4)}_{i,\mathtt{v}}\right)$. So, we need to prove that

$$f^{-1}_{\mathtt{vc}}\left(f_{\mathtt{v}_1\mathtt{c}}\left(\Psi^{(4)}_{\mathtt{c},\mathtt{v}_1} \cap \Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}}\right) + f_{\mathtt{v}_2\mathtt{c}}\left(\Psi^{(4)}_{\mathtt{c},\mathtt{v}_2} \cap \Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}\right)\right) \supseteq \Psi^{(4)}_{\mathtt{c},\mathtt{v}} \cap \Psi^{(1)}_{\mathtt{v},\mathtt{c}}.$$

As the decoder has converged,

$$\begin{aligned}
\Psi^{(4)}_{\mathtt{c},\mathtt{v}} &= f^{-1}_{\mathtt{vc}}\left(f_{\mathtt{v}_1\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}} + f_{\mathtt{v}_2\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}\right),\\
\Psi^{(4)}_{\mathtt{c},\mathtt{v}_1} &= f^{-1}_{\mathtt{v}_1\mathtt{c}}\left(f_{\mathtt{vc}}\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + f_{\mathtt{v}_2\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}\right),\\
\Psi^{(4)}_{\mathtt{c},\mathtt{v}_2} &= f^{-1}_{\mathtt{v}_2\mathtt{c}}\left(f_{\mathtt{v}_1\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}} + f_{\mathtt{vc}}\Psi^{(1)}_{\mathtt{v},\mathtt{c}}\right).
\end{aligned}$$

This implies that

$$f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_1\mathtt{c}}\left(\Psi^{(4)}_{\mathtt{c},\mathtt{v}_1} \cap \Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}}\right) = f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_1\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}} \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_2\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}\right)$$

and

$$f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_2\mathtt{c}}\left(\Psi^{(4)}_{\mathtt{c},\mathtt{v}_2} \cap \Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}\right) = f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_2\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}} \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_1\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}}\right).$$

Let $P = f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_1\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_1,\mathtt{c}}$, $Q = f^{-1}_{\mathtt{vc}} f_{\mathtt{v}_2\mathtt{c}}\Psi^{(1)}_{\mathtt{v}_2,\mathtt{c}}$. Then we need to prove that

$$\left(P \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + Q\right)\right) + \left(Q \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + P\right)\right) \supseteq \Psi^{(1)}_{\mathtt{v},\mathtt{c}} \cap (P+Q). \tag{5}$$

To do this, let $d \in \Psi^{(1)}_{\mathtt{v},\mathtt{c}} \cap (P+Q)$, then $d \in \Psi^{(1)}_{\mathtt{v},\mathtt{c}}$. Also $d = p+q$, where $p \in P, q \in Q$. This implies that $p \in \Psi^{(1)}_{\mathtt{v},\mathtt{c}} + Q$ and $q \in \Psi^{(1)}_{\mathtt{v},\mathtt{c}} + P$. Hence $p \in \left(P \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + Q\right)\right)$ and $q \in \left(Q \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + P\right)\right)$. Hence $p+q \in \left(P \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + Q\right)\right) + \left(Q \cap \left(\Psi^{(1)}_{\mathtt{v},\mathtt{c}} + P\right)\right)$. This proves (5) and also implies that B is a stopping constellation. From Lemma 3.3 it follows that B is the largest stopping constellation contained in C. ■

In the next section we define the residual NBLDPC ensemble resulted by the BP decoder. We compute the design rate and expectation of total number of codewords of the residual ensemble. Then we show how we can derive the residual degree distribution from the fixed points of density evolution for the BP decoder. This will enable us to compute the conditional entropy when the block length tends to infinity.

IV. Residual Degree Distribution and Counting Argument

Assume that the BP decoder has converged. In the previous section we saw that the BP decoder assigns state $B_\mathtt{v}$ to a variable node $\mathtt{v}$, $B_\mathtt{v} \in \mathcal{M}_{\text{ini}}$, $\mathtt{v} \in \mathcal{V}$. We say that the *BP state* of $\mathtt{v}$ is $B_\mathtt{v}$. A variable node $\mathtt{v}$ can only take values belonging to its BP state $B_\mathtt{v}$. Thus in the resulting graph, which we call the *residual graph*, every variable node is characterized by its degree and state. So the *residual degree distribution* on the variable node side is given by $\Omega = \{\Omega_{\mathtt{1}V}\}$, where $\Omega_{\mathtt{1}V}$ denotes the fraction of variable nodes which have degree $\mathtt{1}$ and BP state V. Similarly, we define $\Omega_{\mathtt{1}k}$ to be the fraction of variable nodes which can only take values in a subspace

of dimension k. More precisely,

$$\Omega_{\mathtt{l}k} = \sum_{V:\mathrm{Dim}(V)=k, V\in\mathcal{M}_{\mathrm{ini}}} \Omega_{\mathtt{l}V}.$$

From Lemma 3.1 we know that the BP state assignment is such that every check node satisfies the decoding failure criterion. In order to define the degree distribution of the check node side, we define the set $\mathcal{S}_{\mathtt{r}}$ which consists of all the $\mathtt{r}$-tuples of subspaces which satisfy the decoding failure criterion. More precisely,

$$\mathcal{S}_{\mathtt{r}} = \left\{ (V_1,\ldots,V_{\mathtt{r}}) : V_i \in \mathcal{M}, V_i \subseteq \sum_{j=1, j\neq i}^{\mathtt{r}} V_j, \forall i \in \{1,\ldots,\mathtt{r}\} \right\}. \tag{6}$$

Using the definition of $\mathcal{S}_{\mathtt{r}}$ in (6), we define the residual check node degree distribution $\Phi = \{\Phi_{\mathtt{r}s}\}$. For $s \in \mathcal{S}_{\mathtt{r}}$, $\Phi_{\mathtt{r}s}$ denotes the fraction of check nodes with degree $\mathtt{r}$ and for every such check node $\mathtt{c}$, the state of its neighboring variable nodes when acted on by the corresponding edge labels satisfies $s = \{V_1,\ldots,V_{\mathtt{r}}\}$. More precisely, $V_i = f_{\mathtt{v}_i\mathtt{c}} B_{\mathtt{v}_i}$, where $\mathtt{v}_i \in \mathcal{N}(\mathtt{c})$, $B_{\mathtt{v}_i}$ is the BP state of $\mathtt{v}_i$ and $f_{\mathtt{v}_i\mathtt{c}}$ is the edge label of the edge connecting $\mathtt{v}_i$ and $\mathtt{c}$. We say that the check node $\mathtt{c}$ is of type $(\mathtt{r},s)$ and its i^{th} socket is restricted to the subspace V_i. We do not distinguish between two types of check nodes $(\mathtt{r},s_1)$ and $(\mathtt{r},s_2)$, where s_2 is identical to s_1 up to some permutation. Hence only one arbitrary but fixed representative is included in $\mathcal{S}_{\mathtt{r}}$.

We denote the ensemble of all the residual graphs with block length n and degree distribution (Ω,Φ) by $\mathrm{RESEGL}(n,\Omega,\Phi,m)$ ($\mathrm{RESEGL}(\Omega,\Phi,m)$ in the asymptotic limit). Thus the ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,m)$ has $n\Omega_{\mathtt{l}V}$ number of variable nodes of degree $\mathtt{l}$ which can only take values in subspace V, $V \in \mathcal{M}_{\mathrm{ini}}$. Similarly, there are $n(1-R)\Phi\mathtt{r}s$ check nodes with type $(\mathtt{r},s)$, $s \in \mathcal{S}_{\mathtt{r}}$, where $\mathcal{S}_{\mathtt{r}}$ is defined in (6). Here R is the design rate of the initial ensemble $\mathrm{EGL}(\Lambda,\Gamma,m)$ which results in the residual ensemble. From now on, we say that R is the *initial rate* of the residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,m)$. Note that in the residual graph the total number of variable and check nodes is the same as in the original graph. All the bipartite graphs with state assignments and edge labels such that they have degree distribution Ω and Φ are included in the ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,m)$. There are restrictions on the edge labels and graph connections. For example, the sockets from a variable node restricted to subspace V_1 of dimension k can only connect to a check node socket restricted to subspace V_2 and $\mathrm{Dim}(V_2) = k$. The edge label on such an edge is restricted to those mappings which map subspace V_1 to V_2.

In order to determine the design rate of the residual degree distribution, we need to determine the number of constraints imposed by a check node of type $(\mathtt{r},s)$. In the next lemma we prove that the number of binary constraints imposed by a check node of type $(\mathtt{r},s)$ is $\mathrm{Dim}(\sum_{i=1}^{\mathtt{r}} V_i)$, where $s = (V_1,\ldots,V_{\mathtt{r}})$. We also compute the design rate of the residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,m)$.

Lemma 4.1: Consider a check node of degree $\mathtt{r}$ which represents the parity-check equation

$$\sum_{i=1}^{\mathtt{r}} x_i = 0, \tag{7}$$

where x_i can only take values in the subspace $V_i \subseteq \mathrm{GF}_2^m$. Then the number of binary constraints imposed by such a check node is $\mathrm{Dim}(\sum_{i=1}^{\mathtt{r}} V_i)$. Using this, the design rate of the residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,m)$ with initial rate R is given by

$$R_{\mathrm{res}} = 1 - (1-R)\frac{\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}} \Phi_{\mathtt{r}s}\mathrm{Dim}(s)}{\sum_{\mathtt{l}}\sum_{V\in\mathcal{M}_{\mathrm{ini}}} \Omega_{\mathtt{l}V}\mathrm{Dim}(V)}. \tag{8}$$

Proof: Define the map $f : V_1 \times \cdots \times V_{\mathtt{r}} \mapsto \sum_{i=1}^{\mathtt{r}} V_i$, $f(x_1,\cdots,x_{\mathtt{r}}) \triangleq \sum_{i=1}^{\mathtt{r}} x_i$. Then the number of binary con-

straints imposed by (7) is equal to $\sum_{i=1}^{\mathtt{r}} \mathrm{Dim}\,(V_i) - \mathrm{Dim}\,(\text{kernel } f)$. By rank-nullity theorem [20],

$$\mathrm{Dim}\,(\text{Image } f) + \mathrm{Dim}\,(\text{Kernel } f) = \sum_{i=1}^{\mathtt{r}} \mathrm{Dim}\,(V_i)\,.$$

As $\mathrm{Dim}\,(\text{Image } f) = \mathrm{Dim}\,(\sum_{i=1}^{\mathtt{r}} V_i)$, we obtain that the number of binary constraints imposed by (7) is equal to $\mathrm{Dim}\,(\sum_{i=1}^{\mathtt{r}} V_i)$. This proves the first part of lemma. Using this, we obtain that the number of binary constraints imposed by check nodes of $\mathrm{RESEGL}\,(n,\Omega,\Phi,m)$ is given by

$$n(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}} \Phi_{\mathtt{r}s}\mathrm{Dim}(s).$$

The number of unconstrained bits is given by

$$n\sum_{\mathtt{l}}\sum_{V\in\mathcal{M}_{\text{ini}}} \Omega_{\mathtt{l}V}\mathrm{Dim}\,(V)\,.$$

Hence the design rate of the residual ensemble is

$$R_{\text{res}} = 1-(1-R)\frac{\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}}\Phi_{\mathtt{r}s}\mathrm{Dim}(s)}{\sum_{\mathtt{l}}\sum_{V\in\mathcal{M}_{\text{ini}}}\Omega_{\mathtt{l}V}\mathrm{Dim}\,(V)}.$$

This proves the lemma. ∎

From now onwards, we denote the number of constraints imposed by a check node of type $(\mathtt{r},s)$ with $\mathrm{Dim}(s) = \mathrm{Dim}\,(\sum_{i=1}^{\mathtt{r}} V_i)$, where $s = (V_1,\ldots,V_{\mathtt{r}})$. The design rate R_{res} of the ensemble $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$ is a lower bound on the rate of every code in the ensemble $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$. This is due to the fact that when we compute the design rate, we assume that all the parity-check equations are independent. However, in a code some parity-check equations might be dependent. A criterion for binary LDPC ensembles was derived in [17] which, when satisfied, guarantees that the design rate is equal to the actual rate. Towards generalizing this criterion for residual ensembles $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$, we compute the expectation of N, the total number of codewords in a randomly chosen code from $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$.

Lemma 4.2: Let $N(E_1,\ldots,E_m)$ be the number of codewords in a randomly chosen element of the ensemble $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$ such that, for each such codeword there are E_k edges to which the non-zero values are assigned and which are connected to variable nodes only taking values in the subspace of dimension k, $k\in\{1,\ldots,m\}$. Let N be the total number of codewords in a randomly chosen code from the ensemble $\mathrm{RESEGL}\,(n,\Phi,\Omega,m)$. Then,

$$\mathbb{E}(N) = \sum_{E_1=0}^{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}} \cdots \sum_{E_m=0}^{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}m}} \mathbb{E}\left(N\left(E_1,\ldots,E_m\right)\right), \qquad (9)$$

and

$$\mathbb{E}\left(N\left(E_1,\ldots,E_m\right)\right) = \prod_{k=1}^{m} \frac{\mathrm{coef}\left(\prod_{\mathtt{l}}\left(1+\left(2^k-1\right)u_k^{\mathtt{l}}\right)^{n\Omega_{\mathtt{l}k}}, u_k^{E_k}\right)}{g(m,k)^{E_k}\binom{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}k}}{E_k}} \times \mathrm{coef}\left(\prod_{\mathtt{r}}\prod_{s\in\mathcal{S}_{\mathtt{r}}} q_s\left(v_1,\ldots,v_m\right)^{n(1-r)\Phi_{\mathtt{r}s}}, \prod_{k=1}^{m} v_k^{E_k}\right),$$

where $s\in\mathcal{S}_{\mathtt{r}}$, $s=\{V_1,\ldots,V_{\mathtt{r}}\}$ and $\mathcal{S}_{\mathtt{r}}$ is defined in (6). The function $q_s(v_1,\ldots,v_m)$ is given by

$$q_s(v_1,\ldots,v_m) = \sum_{(i_1\ldots i_k)\subseteq\{1,\ldots,\mathtt{r}\}} q_{i_1\ldots i_k}\prod_{j=1}^{k} v_{\mathrm{Dim}\left(V_{i_j}\right)}. \qquad (10)$$

In (10), $q_{i_1\ldots i_k}$ is the number of permissible edge labels assigned to the edges corresponding to $\left(V_{i_1},\ldots,V_{i_k}\right)$ which will yield a valid codeword when the remaining edges corresponding to $\{V_1,\ldots,V_{\mathtt{r}}\}\backslash\{V_{i_1},\ldots,V_{i_k}\}$ carry the value zero.

Proof: The proof is given in Appendix B. ∎

To compute the quantity $q_{i_1\ldots i_k}$ appearing in (10), we will make use of the following lemma.

Lemma 4.3: Let x_i be a non-zero element of GF_2^m, $M_i\in\mathrm{GL}_2^m$ and $i\in\{1,\ldots,\mathtt{r}\}$. Let

$$\mathcal{Z}_{\mathtt{r}} = \left\{(M_1,\ldots,M_{\mathtt{r}}) : \sum_{i=1}^{\mathtt{r}} M_i x_i = 0\right\},$$

and

$$\mathcal{NZ}_{\mathtt{r}} = \left\{ (M_1, \ldots, M_{\mathtt{r}}) : \sum_{i=1}^{\mathtt{r}} M_i x_i = y \right\},$$

where y is some fixed non-zero element of GF_2^m. Then,

$$\begin{aligned} F_{\mathtt{r}} = |\mathcal{Z}_{\mathtt{r}}| &= \frac{|\mathrm{GL}_2^m|^{\mathtt{r}}}{2^m} \left(1 + \frac{(-1)^{\mathtt{r}}}{(2^m-1)^{\mathtt{r}-1}}\right), \\ G_{\mathtt{r}} = |\mathcal{NZ}_{\mathtt{r}}| &= \frac{|\mathrm{GL}_2^m|^{\mathtt{r}}}{2^m} \left(1 - \frac{(-1)^{\mathtt{r}}}{(2^m-1)^{\mathtt{r}}}\right). \end{aligned}$$

Proof: Note that the sequence $\{F_{\mathtt{r}}\}_{\mathtt{r}=1}^{\infty}$ satisfies the following recursive relation:

$$F_{\mathtt{r}+1} = \frac{|\mathrm{GL}_2^m|}{2^m - 1} \left(|\mathrm{GL}_2^m|^{\mathtt{r}} - F_{\mathtt{r}}\right).$$

This follows from the fact that if we look at a particular set of $\mathtt{r}$ terms, they must not sum to zero in order to make sure that $\mathtt{r}+1$ terms sum to zero. Also, there are $\frac{|\mathrm{GL}_2^m|}{2^m-1}$ elements in GL_2^m which maps a given non-zero element x to another non-zero element y, where $x, y \in \mathrm{GF}_2^m$. By solving the recursion, we get

$$F_{\mathtt{r}} = \frac{|\mathrm{GL}_2^m|^{\mathtt{r}}}{2^m} \left(1 + \frac{(-1)^{\mathtt{r}}}{(2^m-1)^{\mathtt{r}-1}}\right).$$

Similarly, the sequence $\{G_{\mathtt{r}}\}$ satisfies the recursive relation

$$G_{\mathtt{r}+1} = \frac{|\mathrm{GL}_2^m|}{2^m - 1} \left((2^m - 2)\, G_{\mathtt{r}} + F_{\mathtt{r}}\right).$$

The solution of this relationship yields

$$G_{\mathtt{r}} = \frac{|\mathrm{GL}_2^m|^{\mathtt{r}}}{2^m} \left(1 - \frac{(-1)^{\mathtt{r}}}{(2^m-1)^{\mathtt{r}}}\right).$$

■

In the remainder of this section, we show how we can compute the *average residual degree distribution* in the limit of infinite block length. First we show that every element of a residual ensemble has uniform probability conditioned on the event that a random residual graph has the degree distribution of the considered residual ensemble.

Lemma 4.4: Let $\mathrm{RESEGL}(n, \Omega, \Phi, m)$ be a residual ensemble. Conditioned on the event that a random residual graph is an element of $\mathrm{RESEGL}(n, \Omega, \Phi, m)$, it is equally likely to be any element of the ensemble $\mathrm{RESEGL}(n, \Omega, \Phi, m)$.

Proof: The proof is given in Appendix C. ■

We denote the ensemble corresponding to the average residual degree distribution by $\mathrm{RESEGL}(n, \Omega, \Phi, m, \varepsilon)$. We prove the following concentration result on the degree distribution of a random residual graph around the average residual degree distribution.

Lemma 4.5: Let $\mathrm{RESEGL}(n, \Omega, \Phi, m, \varepsilon)$ be the average residual degree distribution of the ensemble $\mathrm{EGL}(\Lambda, \Gamma, m)$ assuming the transmission over the $BEC(\varepsilon)$. The residual degree distribution of a random residual graph G is denoted by $\mathrm{RESEGL}(n, \Omega_G, \Phi_G, m, \varepsilon)$. Then, for any $\delta > 0$,

$$\lim_{n \to \infty} \mathrm{P}\left\{ d\left((\Omega, \Phi), (\Omega_G, \Phi_G)\right) \geq \delta \right\} = 0.$$

The distance $d(\cdot, \cdot)$ is the L_1 distance

$$d\left((\Omega, \Phi), (\tilde{\Omega}, \tilde{\Phi})\right) = \sum_{\mathtt{l}, V} |\Omega_{\mathtt{l}V} - \tilde{\Omega}_{\mathtt{l}V}| + \sum_{\mathtt{r}} \sum_{s \in \mathcal{S}_{\mathtt{r}}} |\Phi_{\mathtt{r}s} - \tilde{\Phi}_{\mathtt{r}s}|, \tag{11}$$

where $V \in \mathcal{M}_{\mathrm{ini}}$.

Proof: The proof is given in Appendix A. ■

Asymptotically, Ω and Φ depend on the probability density of the messages of the BP decoder which can be evaluated by density evolution. The density evolution equations for NBLDPC ensemble are derived in [13]. Following the notation in [13], we denote the fixed point probability of the event that the message from a variable(check) node of degree $\mathtt{l}$ is of dimension i by $\mathrm{P}_v(i, \mathtt{l})$($\mathrm{P}_c(i, \mathtt{l})$). Then

$$\Omega_{\mathtt{l}V} = \frac{\Lambda_{\mathtt{l}}\, \mathrm{P}_v(k, \mathtt{l}+1)}{\binom{m}{k}}, \tag{12}$$

where $V \in \mathcal{M}_{\mathrm{ini}}$ and $\mathrm{Dim}(V) = k$. Note that we use $\mathtt{l}+1$ as we take into account all the incoming messages while

computing the estimate of a symbol. We divide by $\binom{m}{k}$, which is the number of initial subspaces of dimension k, to compute the probability of a specific subspace of dimension k.

Deriving the residual degree distribution on the check node side is less straightforward than on the variable node side. Towards this end, observe that we can determine the BP estimate of a symbol if we know the incoming and the outgoing messages to this symbol on one of its connected edges when the BP decoder has converged. Hence if we know all the incoming messages to a check node, we can determine the final estimates of the symbols connected to this check node. Thus from the fixed points of density evolution we can determine the residual degree distribution on the check node side. More precisely, consider a check node of degree $\mathtt{r}$. Let $V_1,\ldots,V_{\mathtt{r}}$ be the incoming messages which are independently distributed according to $\{\mathrm{P}_v(i)\}_{i=0}^{m}$. Then the state of the check node is

$$\left\{V_1 \cap \left(\sum_{j\neq 1} V_j\right),\ldots,V_{\mathtt{r}} \cap \left(\sum_{j\neq \mathtt{r}} V_j\right)\right\}, \qquad (13)$$

which is an element of $\mathcal{S}_{\mathtt{r}}$, $\mathcal{S}_{\mathtt{r}}$ having been defined in (6). Thus determining Φ amounts to determining all the elements of $\mathcal{S}_{\mathtt{r}}$ for all the check node degrees $\mathtt{r}$ and to compute the probability distribution over $\mathcal{S}_{\mathtt{r}}$. In the next section we show how this can be done for the particular case of $m=2$.

V. Particular case of $m=2$

The explicit characterization of quantities in (13) is still not known for all values of m and seems to be difficult to obtain analytically. In what follows we lead the analysis for the simple case of the smallest value of m greater than 1, that is $m=2$. For $m=2$, we compute the check node distribution and prove the equality between the conditional entropy and the rate of the residual ensemble.

A. Calculating the Check Node Distribution for $m=2$

Note that there are five different subspaces of GF_2^2. There is one each of dimension zero (containing the origin only) and of dimension two (GF_2^2). There are three different subspaces of dimension one which we number in an arbitrary but fixed way. The sum of two different subspaces of dimension one gives the subspace of dimension two. More precisely, let S_1,S_2,S_3 be the subspaces of dimension one and T be the subspace of dimension two. Then,

$$T = S_i + S_j, \quad i,j \in \{1,2,3\}, i \neq j. \qquad (14)$$

We denote by $n_0(n_2)$ the number of subspaces of dimension 0(2) in the state $s \in \mathcal{S}_{\mathtt{r}}$ of a check node. Similarly, the number of different subspaces of dimension one is given by $n_1(i)$, $i \in \{1,2,3\}$ and let

$$n_1 = \sum_{i=1}^{3} n_1(i).$$

Now, we enumerate all the possible states of a check node of degree $\mathtt{r}$ which satisfy the decoding failure criterion and derive their probabilities in terms of the fixed points of density evolution. Recall that if the converged incoming messages to a check node are $\{V_1,\ldots,V_{\mathtt{r}}\}$, its state is given by (13).

1) Every state s for which $n_2 \geq 2$ satisfies the decoding failure criterion. Its probability is given by

$$\Phi_{\mathtt{r}s} = \binom{\mathtt{r}}{n_0,n_1(1),n_1(2),n_1(3)} \mathrm{P}_v(2)^{n_2} \times \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1(1)+n_1(2)+n_1(3)} \mathrm{P}_v(0)^{n_0}. \qquad (15)$$

The equation (15) can be easily understood by observing

that if

$$\exists k \in \{1,\ldots,\mathtt{r}\}, k \neq i, \mathrm{Dim}\,(V_k) = 2,$$

then

$$V_i \cap \left(\sum_{i \neq j} V_j\right) = V_i.$$

So, the socket i is restricted to its corresponding incoming message V_i.

2) When s is such that $n_2 = 1$, for s to satisfy the decoding failure criterion we need $n_1(i) > 0$, $n_1(j) > 0$, and $i \neq j$. In this case the sockets are also restricted to their corresponding incoming messages. So, the probability is the same as given by (15).

3) Every state for which only one of $n_1(1), n_1(2), n_1(3)$ is positive and greater than one with $n_2 = 0$ satisfies the decoding failure criterion. Lets us assume w.l.o.g that $n_1(1) > 1$. The probability of such a state s is given by

$$\begin{aligned}\Phi_{\mathtt{r}s} = &\binom{\mathtt{r}}{1, n_1(1)-1} \mathrm{P}_v(2) \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1(1)-1} \mathrm{P}_v(0)^{\mathtt{r}-n_1(1)-1} \\ &+ 2\binom{\mathtt{r}}{1, n_1(1)} \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1(1)+1} \mathrm{P}_v(0)^{\mathtt{r}-n_1(1)-1} \mathbb{1}_{n_1(1)<\mathtt{r}} \\ &+ \binom{\mathtt{r}}{n_1(1)} \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1(1)} \mathrm{P}_v(0)^{\mathtt{r}-n_1(1)}, \quad (16)\end{aligned}$$

where $\mathbb{1}_{n_1(1)<\mathtt{r}}$ is equal to one if $n_1(1) < \mathtt{r}$ and zero otherwise. In order to understand (16), observe that for $n_2 = 0$ to hold, at most one of the incoming messages can be of dimension two. If there are more than one incoming messages of dimension two, then the state of the check node has $n_2 \geq 2$. The first term on the right hand side of (16) corresponds to the case when one of the incoming message is of dimension two, $n_1(1) - 1$ incoming messages are equal to S_1, and rest of messages are equal to the subspace of dimension zero. By (13), we see that such a combination of messages results in the desired check node state s. The second summation term of (16) corresponds to the case where $n_1(1)$ incoming messages are equal to S_1 and one message is equal to S_2 or S_3. This choice of S_2 or S_3 explains the factor 2 in the second summation term. From (13), we see that this combination of incoming messages corresponds to desired state s. The last term takes care of the case when there are $n_1(1)$ incoming messages equal to S_1 and remaining messages are equal to subspace of dimension zero.

4) Consider a state s for which two different subspaces of dimension one are present at least twice and the remaining subspace of dimension 1 is absent with $n_2 = 0$. It can be seen that s satisfies the decoding failure criterion. Assume w.l.o.g. that $n_1(1) \geq 2, n_1(2) \geq 2$, $n_1(3) = 0$, and $n_2 = 0$. Its probability can be derived by using arguments similar to the derivation of (14) and (16). The probability is given by

$$\Phi_{\mathtt{r}s} = \binom{\mathtt{r}}{n_1(1), n_1(2)} \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1(1)+n_1(2)} \mathrm{P}_v(0)^{\mathtt{r}-n_1(1)-n_1(2)}.$$

5) Every state s for which all the subspaces of dimension one are present at least once i.e. $n_1(i) > 0$, $\forall i \in \{1,2,3\}$ and $n_2 = 0$ satisfies the decoding failure criterion. The probability of such a state is given by

$$\Phi_{\mathtt{r}s} = \binom{\mathtt{r}}{n_1(1), n_1(2), n_1(3)} \left(\frac{\mathrm{P}_v(1)}{3}\right)^{n_1} \mathrm{P}_v(0)^{\mathtt{r}-n_1}. \quad (17)$$

Note that in this case each socket is also restricted to its incoming message. This explains (17).

6) Every state s for which $n_0 = \mathtt{r}$ satisfies the decoding failure criterion. Its probability is given by

$$\begin{aligned}\Phi_{\mathtt{r}s} = &\mathrm{P}_v(0)^{\mathtt{r}} + \mathtt{r}\,\mathrm{P}_v(0)^{\mathtt{r}-1}\left(\mathrm{P}_v(1) + \mathrm{P}_v(2)\right) \\ &+ 3\frac{\mathtt{r}!}{(\mathtt{r}-2)!}\left(\frac{\mathrm{P}_v(1)}{3}\right)^2 \mathrm{P}_v(0)^{\mathtt{r}-2}. \quad (18)\end{aligned}$$

The first term on the right hand side of (18) is obvious as it corresponds to the case when all the incoming

messages have dimension zero. The second term is for the following combination of messages. One of the messages is a subspace of dimension one (two) and the remaining messages are subspaces of dimension zero. As the dimension of the intersection of a subspace of dimension one (two) with a subspace of dimension zero is zero, this combination of incoming messages results in the desired state. The remaining terms are derived in a similar way.

In next subsection we show that under appropriate technical conditions, the average conditional entropy of the transmitted codeword is given by the design rate of the average residual ensemble.

B. Equality between the Conditional Entropy and the Rate

In order to derive a sufficient condition which guarantees that asymptotically almost every code in the residual ensemble has its rate equal to the design rate, we first find the generating function $q_s(v_1,v_2)$ defined in (10). Consider a check node of degree $\mathtt{r}$ with a state s such that there are n_0 sockets corresponding to subspaces of dimension zero, $n_1(i)$ sockets corresponding to subspace i of dimension one, $i\in\{1,2,3\}$ and n_2 sockets corresponding to subspace of dimension two. Then the generating function for this check node type is given by

$$q_s(v_1,v_2)=\sum_{S_1}\left\{\binom{n_2}{i}F_i v_2^i\binom{n_1(1)}{j_1}\binom{n_1(2)}{j_2}\binom{n_1(3)}{j_3}2^{j_1+j_2+j_3}v_1^{j_1+j_2+j_3}\right\}+\sum_{S_2}\left\{\binom{n_2}{i}G_i v_2^i\binom{n_1(1)}{j_1}\binom{n_1(2)}{j_2}\binom{n_1(3)}{j_3}2^{j_1+j_2+j_3}v_1^{j_1+j_2+j_3}\right\}, \quad (19)$$

where F_i, G_i are defined in Lemma 4.3, S_1 corresponds to summation over the terms such that $j_1(1), j_1(2), j_1(3)$ are all even or all odd, and S_2 is the complement of S_1. A simplification of $q_s(v_1,v_2)$ yields

$$q_s(v_1,v_2)=f(v_1)\left(\frac{(1+6v_2)^{n_2}+3(1-2v_2)^{n_2}}{4}\right)+((1+2v_1)^{n_1}-f(v_1))\left(\frac{(1+6v_2)^{n_2}-(1-2v_2)^{n_2}}{4}\right), \quad (20)$$

where

$$f(v_1)=\prod_{i=1}^{3}\frac{(1+2v_1)^{n_1(i)}-(1-2v_1)^{n_1(i)}}{2}+\prod_{i=1}^{3}\frac{(1+2v_1)^{n_1(i)}+(1-2v_1)^{n_1(i)}}{2}.$$

In the following lemma we upper bound the difference between the growth rate of the expectation of the total number of codewords and the design rate of the residual ensemble.

Lemma 5.1: Let N be the total number of codewords of a randomly chosen code from the ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,2)$. Then for $\forall u_1\in[0,\infty)$, $\forall u_2\in[0,\infty)$

$$\lim_{n\to\infty}\frac{\log(\mathbb{E}(N))}{n}-R_{res}\le\theta(u_1,u_2).$$

The function $\theta(u_1,u_2)$ for $u_1\in[0,\infty)$ and $u_2\in[0,\infty)$ is defined as follows.

$$\theta(u_1,u_2)=\sum_{\mathtt{l}}\Omega_{\mathtt{l}1}\log_2\left(\frac{1+u_1^{\mathtt{l}}}{2}\right)+\sum_{\mathtt{l}}\Omega_{\mathtt{l}2}\log_2\left(\frac{1+3u_2^{\mathtt{l}}}{4}\right)+(1-r)\sum_{\mathtt{r}}\sum_{s\in S_{\mathtt{r}}}\Phi_{\mathtt{r}s}\log\left(h_s(u_1,u_2)2^{\mathrm{Dim}(s)}\right)-t_{+1}(1)\log_2(t_{+1}(1))-t_{+2}(1)\log_2(t_{+2}(1)), \quad (21)$$

where

$$h_s(u_1,u_2)=p(u_1)\frac{t_{+2}(u_2)^{n_2}+3t_{-2}(u_2)^{n_2}}{4}+\left(t_{+1}^{n_1}(u_1)-p(u_1)\right)\frac{t_{+2}(u_2)^{n_2}-t_{-2}(u_2)^{n_2}}{4},$$

$$t_{+1}(u_1)=\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\frac{1+u_1^{\mathtt{l}-1}}{1+u_1^{\mathtt{l}}},\quad t_{-1}(u_1)=\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\frac{1-u_1^{\mathtt{l}-1}}{1+u_1^{\mathtt{l}}},$$
$$t_{+2}(u_2)=\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\frac{1+3u_2^{\mathtt{l}-1}}{1+3u_2^{\mathtt{l}}},\quad t_{-2}(u_2)=\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\frac{1-u_2^{\mathtt{l}-1}}{1+3u_2^{\mathtt{l}}},$$

$$p(u_1)=\prod_{i=1}^{3}\frac{t_{+1}(u_1)^{n_1(i)}-t_{-1}(u_1)^{n_1(i)}}{2}+\prod_{i=1}^{3}\frac{t_{+1}(u_1)^{n_1(i)}+t_{-1}(u_1)^{n_1(i)}}{2}.$$

Proof: Let $E_1=ne_1\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}$, $E_2=ne_2\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}$, $e_1\in[0,1]$ and $e_2\in[0,1]$. By (9),

$$\lim_{n\to\infty}\frac{\log(\mathbb{E}(N))}{n}=\sup_{e_1,e_2}\lim_{n\to\infty}\frac{\log(\mathbb{E}(N(e_1,e_2)))}{n}.$$

By using Lemma 4.2 and the Hayman approximation of [21] for the coef term, we get

$$\phi(e_1,e_2)=\lim_{n\to\infty}\frac{\log_2(\mathbb{E}[N(e_1,e_2)])}{n}:=\inf_{u_1,u_2,v_1,v_2}\phi_1(e_1,e_2,v_1,v_2,u_1,u_2),$$

where

$$\begin{aligned}\phi_1(e_1,e_2,v_1,v_2,u_1,u_2)&=\textstyle\sum_{\mathtt{l}}\Omega_{\mathtt{l}1}\log_2(1+u_1^{\mathtt{l}})\\&\quad-e_1\log_2(u_1)\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}+\sum_{\mathtt{l}}\Omega_{\mathtt{l}2}\log_2\left(1+3u_2^{\mathtt{l}}\right)\\&\quad-e_2\log_2(u_2)\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}+(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}}\Phi_{\mathtt{r}s}\log(h_s(v_1,v_2))\\&\quad-e_1(\log_2(v_1)+1)\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}-e_2(\log_2(v_2)+\log_2(6))\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\\&\quad-\left(\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\right)h(e_1)-\left(\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\right)h(e_2).\end{aligned}$$

Hence we would like to compute $\sup_{e_1,e_2}\phi(e_1,e_2)$, where $e_1\in[0,1]$ and $e_2\in[0,1]$. Thus,

$$\sup_{e_1,e_2}\phi(e_1,e_2)=\sup_{e_1,e_2}\inf_{v_1,v_2,u_1,u_2}\phi_1(e_1,e_2,v_1,v_2,u_1,u_2).\tag{22}$$

We find an upper bound on $\sup_{e_1,e_2}\phi(e_1,e_2)$. Towards this end, first take the derivative of $\phi_1(e_1,e_2,u_1,u_2,v_1,v_2)$ with respect to e_1 and e_2 and equate it to zero. This gives

$$e_1=\frac{2u_1v_1}{1+2u_1v_1},\quad e_2=\frac{6u_2v_2}{1+6u_2v_2}.$$

We substitute this in the expression for $\phi_1(e_1,e_2,u_1,u_2,v_1,v_2)$. After that we take the derivative with respect to u_1, u_2 and equate it to zero. Solving for v_1 and v_2, we obtain

$$v_1(u_1)=\frac{1}{2}\frac{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\frac{u_1^{\mathtt{l}-1}}{1+u_1^{\mathtt{l}}}}{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\frac{1}{1+u_1^{\mathtt{l}}}},\quad v_2(u_2)=\frac{1}{6}\frac{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\frac{3u_2^{\mathtt{l}-1}}{1+3u_2^{\mathtt{l}}}}{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\frac{1}{1+3u_2^{\mathtt{l}}}}.$$

Substituting the expression for $v_1(u_1)$ and $v_2(u_2)$ in the expression for e_1 and e_2, we get

$$e_1(u_1)=\frac{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}\frac{u_1^{\mathtt{l}}}{1+u_1^{\mathtt{l}}}}{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}},\quad e_2(v)=\frac{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}\frac{3u_2^{\mathtt{l}}}{1+3u_2^{\mathtt{l}}}}{\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}}.$$

Let $\theta(u_1,u_2)=\phi_1(e_1(u_1),e_2(u_2),v_1(u_1),v_2(u_2),u_1,u_2)-R_{res}$. By rearranging terms, we get the desired expression for $\theta(u_1,u_2)$. ■

Now, we give the criterion which, when satisfied, yields that almost every code in the residual ensemble has its rate equal to the design rate.

Lemma 5.2: Let G be a code chosen uniformly at random from the residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,2)$ and let R_G be its rate. If the function $\theta(u_1,u_2)$ defined in (21) satisfies

$$\theta(u_1,u_2)\le 0,\quad \forall u_1\in[0,\infty),u_2\in[0,\infty),$$

then there exists $B>0$ and a positive integer n_0 such that, for any $\eta>0$,

$$\mathrm{P}(|R_G-R_{res}|>\eta)\le e^{-Bn\eta}.$$

In addition, there exists $C>0$ such that, for $n>n_0$,

$$\mathbb{E}(|R_G-R_{res}|)\le C\frac{\log(n)}{n},$$

where R_{res} is the design rate of the residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,2)$. Thus, almost every code in the

residual ensemble has its rate equal to its design rate.

Proof: This follows from the same arguments as that of Lemma 7 of [17]. ∎

We now prove some properties of the function $\theta(u_1,u_2)$ defined in (21), which will help us in showing that the conditional entropy of NBLDPC ensemble $\mathrm{EGL}(\Lambda,\Gamma,2)$ is given by the design rate of the average residual ensemble. By Lemma 5.2, the maximum of the function $\theta(u_1,u_2)$ determines if the rate of of the residual ensemble is equal to its design rate. We prove that $\theta(u_1,u_2)$ attains its maximum in the unit square $[0,1]^2$.

Lemma 5.3: Consider the ensemble $\mathrm{RESEGL}(\Omega,\Phi,2)$. Then the function $\theta(u_1,u_2)$ defined in (21) attains its maximum inside the unit square. The function $\theta(u_1,u_2)$ depends smoothly on its residual degree distribution. More precisely, there exist constants $B_1,B_2,B_3>0$ such that, for any two residual degree distributions $\mathrm{RESEGL}(\Omega,\Phi,2)$ and $\mathrm{RESEGL}\left(\tilde{\Omega},\tilde{\Phi},2\right)$, having the same maximum left degree and the same maximum check node degree, the corresponding functions $\theta(u_1,u_2)$ and $\tilde{\theta}(u_1,u_2)$ satisfy for $u_1,u_2\in[0,1]$,

$$\begin{aligned}&\left|\theta(u_1,u_2)-\tilde{\theta}(u_1,u_2)\right|\leq d\left((\Omega,\Phi),\left(\tilde{\Omega},\tilde{\Phi}\right)\right)\\&\left(B_1(1-u_1)^2+B_2(1-u_1)(1-u_2)+B_3(1-u_2)^2\right),\end{aligned}\tag{23}$$

where $d(\cdot,\cdot)$ is the L_1 distance defined in (11).

Proof: We refer the reader to Appendix D for the proof. ∎

We now show the equality between the design rate of the residual ensemble and the conditional entropy of the transmitted ensemble.

Theorem 5.1: Consider transmission over $\mathrm{BEC}(\varepsilon)$. Let G be a code chosen uniformly at random from the NBLDPC ensemble $\mathrm{EGL}(n,\Lambda,\Gamma,2)$ and let $H_G(X|Y)$ be its conditional entropy [1]. Let $\mathrm{RESEGL}(n,\Omega,\Phi,2,\varepsilon)$ be the average residual ensemble of $\mathrm{EGL}(n,\Lambda,\Gamma,2)$. Consider the corresponding function $\theta(u_1,u_2)$ for $\mathrm{RESEGL}(n,\Omega,\Phi,2,\varepsilon)$ defined in (21). Assume that $(u_1,u_2)=(1,1)$ is a unique global maximum of $\theta(U_1,u_2)$ for $u_1\in[0,\infty),u_2\in[0,\infty)$, with

$$\frac{\partial^2\theta(u_1,u_2)}{\partial u_1^2}\frac{\partial^2\theta(u_1,u_2)}{\partial u_2^2}-\left(\frac{\partial^2\theta(u_1,u_2)}{\partial u_1\partial u_2}\right)^2\Bigg|_{u_1=1,u_2=1}>0,\tag{24}$$

$$\frac{\partial^2\theta(u_1,u_2)}{\partial u_1^2}\Bigg|_{u_1=1,u_2=1}<0.\tag{25}$$

Then

$$\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left(H_G(X|Y)\right)=R_{res},$$

where R_{res} is the design rate of the average residual ensemble $\mathrm{RESEGL}(n,\Omega,\Phi,2,\varepsilon)$.

Proof: This theorem is a straightforward generalization of [17, Thm. 10]. So the proof is very similar to that of [17, Thm. 10]. Together, (24) and (25) guarantee that the point $(u_1,u_2)=(1,1)$ is a maximum and imply that

$$\frac{\partial^2\theta(u_1,u_2)}{\partial u_2^2}\Bigg|_{u_1=1,u_2=1}<0.\tag{26}$$

By (25, 26), the assumption that the point $(1,1)$ is a global maximum of $\theta(u_1,u_2)$ and

$$\theta(1,1)=0,\quad \frac{\partial\theta(u_1,u_2)}{\partial u_1}\Bigg|_{u_1=1,u_2=1}=0,$$

$$\frac{\partial\theta(u_1,u_2)}{\partial u_2}\Bigg|_{u_1=1,u_2=1}=0,$$

there exist constants $\delta,C_1,C_3>0$ and C_2 such that $\forall u_1\in$

[1] The notation of the conditional entropy has been defined in Section II.

$[0,1], u_2 \in [0,1]$,

$$\theta(u_1,u_2) \leq -\delta\left(C_1(1-u_1)^2 + C_2(1-u_1)(1-u_2) + C_3(1-u_2)^2\right),$$

and

$$\left(C_1(1-u_1)^2 + C_2(1-u_1)(1-u_2) + C_3(1-u_2)^2\right) > 0, \quad \forall u_1 \in [0,1), u_2 \in [0,1).$$

Using Lemma 5.3, we observe that there exist $\eta > 0$ such that for every residual degree distribution pair $(\tilde{\Omega}, \tilde{\Phi})$, which satisfies $d\left((\Omega,\Phi),(\tilde{\Omega},\tilde{\Phi})\right) \leq \eta$, its corresponding function $\tilde{\theta}(u_1,u_2)$ is upper bounded in the interval $u_1 \in [0,1], u_2 \in [0,1]$ by

$$\tilde{\theta}(u_1,u_2) \leq -\frac{\delta}{2}\left(C_1(1-u_1)^2 + C_2(1-u_1)(1-u_2) + C_3(1-u_2)^2\right).$$

Thus Lemma 5.2 is applicable to $\text{RESEGL}\left(n,\tilde{\Omega},\tilde{\Phi},2\right)$, and the design rate $\tilde{R}_{res}$ of $\text{RESEGL}\left(n,\tilde{\Omega},\tilde{\Phi},2\right)$ is equal to its average rate $\tilde{R}$. Let $Q(\eta)$ be the set of residual ensembles whose degree distribution pair $\left(\tilde{\Omega},\tilde{\Phi}\right)$ satisfies $d\left((\Omega,\Phi),(\tilde{\Omega},\tilde{\Phi})\right) \leq \eta$. Let $\text{P}_\varepsilon\left(\tilde{\mathcal{R}}\right)$ be the probability that a random residual graph belongs to the residual ensemble $\tilde{\mathcal{R}} = \text{RESEGL}\left(n,\tilde{\Omega},\tilde{\Phi},2\right)$. Then the conditional entropy of the ensemble $\text{RESEGL}(n,\Lambda,\Gamma,2)$ is given by

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left[H_G(X|Y)\right] &= \sum_{\tilde{\mathcal{R}}} \text{P}_\varepsilon\left(\tilde{\mathcal{R}}\right)\tilde{R}, \\ &= \sum_{\tilde{\mathcal{R}}\in Q(\eta)} \text{P}_\varepsilon\left(\tilde{\mathcal{R}}\right)\tilde{R} + \gamma(n,\eta).\end{aligned}$$

By Lemma 4.5 and the fact that $\tilde{R} \leq 1$, the term $\gamma(n,\eta)$ satisfies

$$\lim_{n\to\infty}\gamma(n,\eta) = 0.$$

Now,

$$\begin{aligned}\left|\frac{1}{n}\mathbb{E}\left[H_G(X|Y)\right] - R_{res}\right| &\leq \sum_{\tilde{\mathcal{R}}\in Q(\eta)} \text{P}_\varepsilon\left(\tilde{\mathcal{R}}\right)\left|\tilde{R} - R_{res}\right| + \gamma'(n,\eta), \\ &\leq \sum_{\tilde{\mathcal{R}}\in Q(\eta)} \text{P}_\varepsilon\left(\tilde{\mathcal{R}}\right)\left|\tilde{R}_{res} - R_{res}\right| + \gamma'(n,\eta),\end{aligned}$$

where by Lemma 5.2

$$\lim_{n\to\infty}\gamma'(n,\eta) = 0.$$

Note that there exists a constant $C > 0$ such that

$$\left|\tilde{R}_{res} - R_{res}\right| \leq C d\left(\left(\tilde{\Omega},\tilde{\Phi}\right),(\Omega,\Phi)\right).$$

This implies that

$$\lim_{n\to\infty}\left|\frac{1}{n}\mathbb{E}\left[H_G(X|Y)\right] - R_{res}\right| \leq C\eta.$$

As we can chose η to be arbitrarily small, the proof is completed. ■

Remark 1: We note that the design rate of the residual ensemble is always a lower bound on the normalized average conditional entropy of the code. This is because of the fact that the design rate is always a lower bound on the actual rate of a code.

In the next subsection we give an example to show how we can compute the MAP threshold from the computation of conditional entropy.

C. Example

Consider the ensemble $\text{EGL}(\lambda,\rho,2)$, where $\lambda(x) = 0.51x + 0.49x^2$ and $\rho(x) = x^2$. Its BP threshold is equal to $\varepsilon^{\text{BP}} = 0.758$. For $\varepsilon = 0.789$, the design rate of the average residual ensemble becomes zero. Thus from the remark following the proof of Theorem 5.1, $\varepsilon = 0.789$ is an upper bound on the MAP threshold. However, for $\varepsilon \geq 0.789$ the function $\theta(u_1,u_2)$ corresponding to the

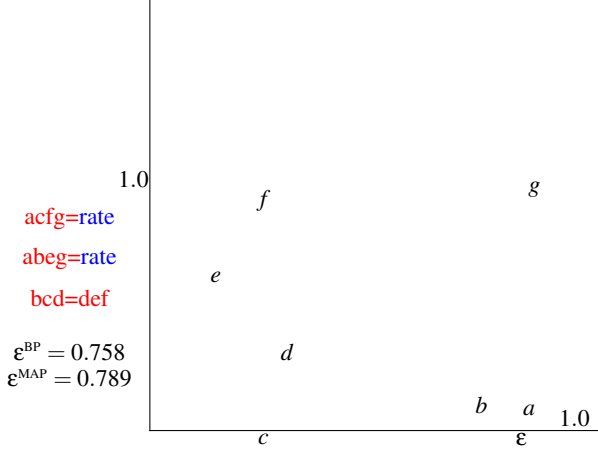

Fig. 1: *EBP curve for EGL* $(0.51x + 0.49x^2, x^2, 2)$. *The MAP threshold is* 0.789 *as computed by the Maxwell construction. The point* $\varepsilon = 0.789$ *corresponds to the erasure probability such that the areas bcd and def are equal.*

average residual degree distribution is non-positive for $u_1, u_2 \in [0,1]$. Also, the assumptions of Theorem 5.1 are satisfied by $\theta(u_1, u_2)$ for $\varepsilon \geq 0.789$. So, the rate of the average residual ensemble is equal to its design rate for $\varepsilon \geq 0.789$ which is equal to the normalized conditional entropy of the ensemble $\mathrm{EGL}(\lambda, \rho, 2)$. The design rate of the average residual ensemble becomes zero for $\varepsilon = 0.789$ and so does the average conditional entropy. This implies that the MAP threshold is equal to $\varepsilon = 0.789$ and the upper bound on the MAP threshold obtained via the design rate is tight. The MAP threshold of the corresponding binary ensemble is given by 0.78.

There is another possible approach of computing the MAP threshold, which is via the Extended-Belief-Propagation Generalized-EXIT (EBP GEXIT) function [17]. In order to define the EBP GEXIT function, let us write the check node side density evolution map as

$$\mathrm{P}_c^{(l)} = G_c\left(\mathrm{P}_v^{(l)}\right),$$

where $\mathrm{P}_c^{(l)}(\mathrm{P}_v^{(l)})$ is the probability distribution of the dimension of the message emanating from the check (variable) node side in the l^{th} iteration. Similarly, we write the density evolution map on the variable node side as

$$\mathrm{P}_v^{(l+1)} = G_v\left(\varepsilon, \mathrm{P}_c^{(l)}\right).$$

Thus the density evolution recursion can be written as

$$\mathrm{P}_v^{(l+1)} = G(\varepsilon, \mathrm{P}_v^{(l)}) = G_v\left(\varepsilon, G_c\left(\mathrm{P}_v^{(l)}\right)\right). \tag{27}$$

We also define another map corresponding to the variable node side, where we do not take into account the channel observation of a variable node. But, we take into account all the incoming messages to this variable node. This corresponds to estimating the variable node from extrinsic observations. Let us denote the resulting distribution as $\mathrm{P}_{v,\text{ext}}^{l}$ and the corresponding map by $G_{v,\text{ext}}$. Then,

$$\mathrm{P}_{v,\text{ext}}^{l} = G_{v,\text{ext}}\left(\mathrm{P}_c^{(l)}\right).$$

Now, the EBP GEXIT function is a parametric function of the fixed point pairs $(\varepsilon_h, \mathrm{P}_v^h)$ of the density evolution map given in (27), i.e., $\mathrm{P}_v^h = G\left(\varepsilon_h, \mathrm{P}_v^h\right)$, where $h \in [0,1]$ is the normalized entropy of P_v^h, i.e.,

$$\frac{1}{2}\sum_{i=0}^{2} i\,\mathrm{P}_v^h(i) = h.$$

For the EBP GEXIT function, the x-coordinate is ε_h and the y-coordinate is h_{ext}, where h_{ext} is the BP extrinsic entropy of a bit, i.e.,

$$h_{\text{ext}} = \frac{\mathrm{P}_{v,\text{ext}}^h(1)}{3} + \frac{\varepsilon_h\,\mathrm{P}_{v,\text{ext}}^h(1)}{3} + \mathrm{P}_{v,\text{ext}}^h(2), \tag{28}$$

where $\mathrm{P}_{v,\text{ext}}^h = G_{v,\text{ext}}\left(G_c(\mathrm{P}_v^h)\right)$. Note that the first term of (28) takes care of the fact that among the three subspaces of dimension one, the value of a given bit is completely erased in one of them. The second term corresponds to the subspace of dimension one in which both the bits take the same value. So, for one to be erased another one should also be erased. The last term takes care of the subspace of dimension two.

When we compute the EBP GEXIT function for this ensemble and apply the Maxwell construction of [17], we get the MAP threshold equal to $\varepsilon^{\text{MAP}} = 0.789$ (Fig. 1)

which is equal to the actual MAP threshold. Thus the Maxwell construction seems to hold in the setting of NBLDPC ensembles. In brief, the Maxwell construction computes the MAP threshold by finding the erasure probability such that the areas def and bcd, shown in Fig. 1, are equal.

VI. Conclusion

We have generalized the concepts of peeling decoder and stopping constellations for NBLDPC codes. Then we showed that both the peeling and BP decoder get stuck in the largest stopping constellation contained in the channel state assignment.

We defined the average residual degree distribution for NBLDPC ensembles and showed how it can be computed for $m = 2$. For the particular case of $m = 2$, we generalized the criterion of [17] which, when satisfied, yields that the actual rate of the residual ensemble is equal to its design rate. This enabled us to compute the conditional entropy and consequently the MAP threshold of the NBLDPC ensemble. There are two main difficulties in generalizing these results to larger values of m. Firstly, an efficient method is required to enumerate all the check node states satisfying the decoding failure criterion and compute their probabilities. Secondly, in order to generalize Theorem 5.1, it is not known how the global maximum of the corresponding function $\theta(u_1, \cdots, u_m)$ can be found.

We presented an observation in Section V-B that the Maxwell construction of [17] seems to hold for $m = 2$. It will be of interest to investigate if the Maxwell construction holds in full generality and prove it analytically as has been done in the binary setting [17].

Acknowledgment

The authors would like to thank Rüdiger Urbanke for helpful comments. We are grateful to anonymous reviewers for very useful and important remarks which have helped in substantially improving the quality of the paper. We thank Igal Sason for careful handling of the paper.

References

[1] V. Rathi, "Conditional entropy of non-binary LDPC codes over the BEC," in *Proc. of the IEEE Int. Symposium on Inform. Theory*, Toronto, Canada, 2008.

[2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, Massachusetts: M.I.T. Press, 1963.

[3] M. Luby and M. Mitzenmacher, "Verification codes," in *Proc. 40th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2002.

[4] A. Bennatan and D. Burshtein, "On the application of LDPC codes to arbitrary discrete-memoryless channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.

[5] D. Sridhara and T. E. Fuja, "LDPC codes over rings for PSK modulation," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3209–3220, May 2005.

[6] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.

[7] M. Flanagan, V. Skachek, E. Byrne, and M. Greferath, "Linear-programming decoding of non-binary linear codes," *IEEE Trans. Inform. Theory*, vol. 55, no. 9, pp. 4134–4154, Sept. 2009.

[8] E. Hof, I. Sason, and S. Shamai, "Performance bounds for non-binary linear block codes over memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 469–477, Mar. 2009.

[9] G. Como and F. Fagnani, "Average spectra and minimum distances of low density parity check codes over abelian groups," *SIAM J. Discr. Math.*, vol. 23, pp. 19–53, 2008.

[10] ——, "The capacity of finite abelian group codes over memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2037–2054, May 2009.

[11] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, jun 1998.

[12] X. Hu, "Low-delay low-complexity error-correcting codes on sparse graphs," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2002.

[13] V. Rathi and R. Urbanke, "Density evolution, threshold and the stability condition for non-binary LDPC codes," *Special issue on Capacity Achieving Codes, IEE Communication Proceedings*, vol. 152, no. 6, pp. 1069–1074, 2005.

[14] C. Poulliat, M. Fossorier, and D. Declercq, "Using binary images of nonbinary LDPC codes to improve overall performance," in *Proc. of Int. Symp. on Turbo-Codes*, Munich, Germany, 2006.

[15] I. Andriyanova and J. P. Tillich, "A family of non-binary TLDPC codes: density evolution, convergence and thresholds," in *Proc. of the IEEE Int. Symposium on Inform. Theory*, Nice, France, July 2007.

[16] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[17] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell's construction: The hidden bridge between maximum-likelihood and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5277–5307, 2008.

[18] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[19] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[20] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[21] E. A. Bender and L. B. Richmond, "Central and local limit theorems applied to asymptotic enumeration II: multivariate generating functions," *J. Combin. Theory*, vol. A 34, no. 3, pp. 255–265, 1983.

[22] N. C. Wormald, "Differential equations for random processes and random graphs," *Ann. Appl. Probab.*, vol. 5, pp. 1217–1235, 1995.

[23] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[24] J. C.H. Edwards, *Advanced Calculus of Several Variables*. Dover publications, 1994.

APPENDIX A
PROOF OF LEMMA 4.5

Proof: Consider a code G chosen randomly from the ensemble $\mathrm{EGL}(n,\Lambda,\Gamma,m)$. We know from Lemma 3.1 and Theorem 3.1 that both the BP and the peeling decoder results in the same residual graph. Let us denote the residual degree distribution of the graph G by $\{\Omega_G,\Phi_G\}$. We prove the concentration of the residual degree distribution by analyzing the peeling decoder. To simplify analysis, we define the peeling decoder in the following way. We pick at random an active check node and update a subspace of the largest dimension. We assume an arbitrary but fixed ordering if there are more than one subspaces of the largest dimension. Note that in one round of the peeling decoder, improving the state dimension of a variable node by i can be seen as i rounds of the decoding where each of them gives an improvement only by 1 dimension. Thus, for simplicity of analysis, we suppose that at each round of the peeling decoder we improve the dimension only by 1.

We use the Wormald's approach [22] to show the concentration of the residual degree distribution. The Wormald's approach was used in [18] to analyze the peeling decoder for binary LDPC codes. We will follow the exposition given in [23, Appendix C.4]. We think of each round of peeling decoder as one time step. We show that the residual degree distribution at any time $t \leq t_S - \eta n$ will be close to the average, where t_S is the time at which the peeling decoder stops and η is any positive constant which can be chosen arbitrarily small. We then show that for time t, $t_S - \eta n \leq t \leq t_S$, the residual degree distribution does not change "significantly". This will prove the concentration of the final residual degree distribution.

To prove the concentration for $t \leq t_S - \eta n$, let us verify that the necessary conditions of the Wormald's method are satisfied [23, Thm. C.28, pp. 491]. To do this, we write down the average change in residual degree distribution for one step of the peeling decoder. At time t, let $W_{\mathtt{l}k}(t)$ denote the number of variable nodes of degree $\mathtt{l}$ with the state dimension k. Let $P_{\mathtt{r}s}(t)$ denote the number of check nodes of degree $\mathtt{r}$ and state $s=(V'_1,\ldots,V'_{\mathtt{r}})$. We denote the set of active check nodes by $\mathcal{A}_{\mathtt{c}}$. Among the set $\mathcal{A}_{\mathtt{c}}$, the subset of check nodes which update a subspace of dimension k is denoted by

$\mathcal{A}_{\mathtt{c}}(k)$ and corresponding to state s is denoted by $\mathcal{A}_{\mathtt{c}}(s)$. To simplify notation, we do not include the time t in $\mathcal{A}_{\mathtt{c}}, \mathcal{A}_{\mathtt{c}}(k)$, and $\mathcal{A}_{\mathtt{c}}(s)$.

The average change in the number of variable nodes of degree $\mathtt{l}$ of the state dimension k is given by

$$\mathbb{E}[W_{\mathtt{l}0}(t+1) - W_{\mathtt{l}0}(t) | W(t), P(t)] = \frac{|\mathcal{A}_{\mathtt{c}}(1)|}{|\mathcal{A}_{\mathtt{c}}|} \frac{\mathtt{l} W_{\mathtt{l}1}(t)}{\sum_d dW_{d1}(t)}, \tag{29}$$

$$\mathbb{E}[W_{\mathtt{l}k}(t+1) - W_{\mathtt{l}k}(t) | W(t), P(t)] = -\frac{|\mathcal{A}_{\mathtt{c}}(k)|}{|\mathcal{A}_{\mathtt{c}}|} \frac{\mathtt{l} W_{\mathtt{l}k}(t)}{\sum_d dW_{dk}(t)} + \frac{|\mathcal{A}_{\mathtt{c}}(k+1)|}{|\mathcal{A}_{\mathtt{c}}|} \frac{\mathtt{l} W_{\mathtt{l},k+1}(t)}{\sum_d dW_{d,k+1}(t)}, \ 1 \le k \le m-1, \tag{30}$$

$$\mathbb{E}[W_{\mathtt{l}m}(t+1) - W_{\mathtt{l}m}(t) | W(t), P(t)] = -\frac{|\mathcal{A}_{\mathtt{c}}(m)|}{|\mathcal{A}_{\mathtt{c}}|} \frac{\mathtt{l} W_{\mathtt{l}m}(t)}{\sum_d dW_{dm}(t)}. \tag{31}$$

The first term in (30) is the average number of variable nodes of initial dimension k, improving their dimension at step $t+1$, and the second term is the average number of variable nodes improving their dimension from $k+1$ to k. In (29) and (31), which represent cases $k=0$ and $k=m$, we have only one of these terms.

After update at the variable node side, we update the state of check nodes connected to the updated variable node. To describe the changes at the check node side, we denote the average change in the number of check nodes of state s, connected to the variable node of interest, given that the chosen active check node has changed its state from s_1 to s_2 by $D(s|s_1 \to s_2)$. More precisely, $D(s|s_1 \to s_2)$ is defined as

$$D(s|s_1 \to s_2) = \mathbb{E}[P_{\mathtt{r}s}(t+1) - P_{\mathtt{r}s}(t) | W(t), P(t), s_1 \to s_2],$$

and can be calculated as follows

$$D(s|s_1 \to s_2) = \frac{\sum_{\mathtt{l}} (\mathtt{l}-1) W_{\mathtt{l},k+1}(t)}{\sum_d W_{d,k+1}(t)} \left(\frac{\sum_{r,s':s' \neq s} \tilde{E}_{\mathtt{r},s'}(t)}{\sum_d dW_{d,k+1}(t)} - \frac{E_{\mathtt{r},s}(t)}{\sum_d dW_{d,k+1}(t)} \right) + O(1/n), \tag{32}$$

where $\tilde{E}_{\mathtt{r},s'}(t)$ denotes the number of sockets of dimension $k+1$ connected to check nodes of state s' of degree $\mathtt{r}$ which change their dimension to k when the corresponding check nodes change their state to s. Similarly, $E_{\mathtt{r},s}(t)$ denotes the number of sockets of dimension k connected to check nodes of state s of degree $\mathtt{r}$ which change their dimension to $k-1$ when the corresponding check nodes change their state s to some other state. Then, the first term in (32) is the expectation of the number of edges going out of the variable node under consideration. The first term in the brackets is the average number of check nodes changing their state from s' to s when the variable node changes the dimension from $k+1$ to k. The second term in the brackets is the average number of check nodes changing their state from s to some other state.

We compute $\mathbb{E}[P_{\mathtt{r}s}(t+1) - P_{\mathtt{r}s}(t) | W(t), P(t)]$ by conditioning on the following three events.

1) The initial state of the chosen active check node is s and is changed to another state.
2) The initial state is s' and is changed to s.
3) the initial and final states are not equal to s.

Then

$$\mathbb{E}[P_{\mathtt{r}s}(t+1) - P_{\mathtt{r}s}(t) | W(t), P(t)] = \frac{|\mathcal{A}_c(s)|}{|\mathcal{A}_c|}(-1 + D(s|s \to s')) + \frac{|\mathcal{A}_c(\tilde{S})|}{|\mathcal{A}_c|}(1 + D(s|\tilde{S} \to s)) + \frac{|\mathcal{A}_c(\bar{S})|}{|\mathcal{A}_c|} D(s|\bar{S} \to \bar{S}') + O(1/n), \tag{33}$$

where s' is the resulting state corresponding to changing s, $\tilde{S}$ is the set of active check nodes which result in s, $\bar{S}$ is the set of active check node states other than s and $\tilde{S}$, and $\bar{S}'$ is the resulting set of check node states corresponding to changing $\bar{S}$. The ratios in all the three terms correspond to the probability of choosing the desired check node state. The -1 in the first term is to take care of the fact that we are updating the check node

state s. The 1 in the second term is present due to the fact that changing a state in $\tilde{S}$ results in s. The other terms are self explanatory by definition.

We need to verify that the conditions of Theorem $C.28$ of [23] holds. The first condition requires that the change in the number of variable (check) nodes of a given state is uniformly bounded. This is true as:

$$\max\{\max_{\mathtt{l},k}|W_{\mathtt{l}k}(t+1)-W_{\mathtt{l}k}(t)|,\max_{\mathtt{r},s}|P_{\mathtt{r}s}(t+1)-P_{\mathtt{r}s}(t)|\} \leq \mathtt{l}_{\max}.$$

The second condition corresponding to the expected change is verified by (29-31, 33). Moreover, the form of (29-31, 33) is very similar to those in the binary case given in [23, Thm. 3.106]. So, the proof of the the Lipschitz continuity and initial concentration are the same. Therefore, all the necessary conditions of [23, Thm. C.28] are satisfied. This results in the following concentration result on the degree distribution $\{\Omega_{G(t)},\Phi_{G(t)}\}$ for $0\leq t\leq t_S-\eta n$ and $\delta>0$

$$\lim_{n\to\infty}\mathrm{P}\left(d\left((\Omega(t),\Phi(t)),(\Omega_{G(t)},\Phi_{G(t)})\right)\geq\delta\right)=0, \quad (34)$$

where $(\Omega(t),\Phi(t)$ is the average degree distribution at time t. Recall that the average residual degree distribution at time t_S is (Ω,Φ) and that of the code G is (Ω_G,Φ_G). By the triangular inequality on the distance, we get the following upper bound:

$$\begin{aligned}&\mathrm{P}\left(d\left((\Omega,\Phi),(\Omega_G,\Phi_G)\right)\geq\delta\right)\leq\\&\quad\mathrm{P}\left(d\left((\Omega,\Phi),(\Omega(t_S-\eta n),\Phi(t_S-\eta n))\right)\geq\delta\right)+\\&\mathrm{P}\left(d\left((\Omega(t_S-\eta n),\Phi(t_S-\eta n)),(\Omega_{G(t_S-\eta n)},\Phi_{G(t_S-\eta n)})\right)\geq\delta\right)+\\&\quad\mathrm{P}\left(d\left((\Omega_{G(t_S-\eta n)},\Phi_{G(t_S-\eta n)}),(\Omega_G,\Phi_G)\right)\geq\delta\right). \quad (35)\end{aligned}$$

Note that

$$\begin{aligned}d\left((\Omega_{G(t_S-\eta n)},\Phi_{G(t_S-\eta n)}),(\Omega_G,\Phi_G)\right) &\leq c\eta, \quad (36)\\ d\left((\Omega(t_S-\eta n),\Phi(t_S-\eta n)),(\Omega,\Phi)\right) &\leq c\eta. \quad (37)\end{aligned}$$

By choosing η sufficiently small in (34), we can make the RHS of (35) zero when n tends to infinity. Hence we obtain the desired concentration of the residual degree distribution. ■

APPENDIX B
PROOF OF LEMMA 4.2

Proof: The proof of (9) is straight forward. Now,

$$\begin{aligned}\mathbb{E}\left[N\left(E_1,\ldots,E_m\right)\right]&=\sum_{w\in W(E_1,\ldots,E_m)}\mathrm{P}\left(w\text{ is a codeword}\right)\\&=\left|W\left(E_1,\ldots,E_m\right)\right|\mathrm{P}\left(w\text{ is a codeword}\right). \quad (38)\end{aligned}$$

Here $W\left(E_1,\ldots,E_m\right)$ denotes the set of words which will yield E_k edges to which the non-zero values are assigned and which are connected to variable nodes taking values only in a subspace of dimension k, $k\in\{1,\ldots,m\}$. Now,

$$\left|W\left(E_1,\ldots,E_m\right)\right|=\prod_{k=1}^{m}\mathrm{coef}\left(\prod_{\mathtt{l}}\left(1+\left(2^k-1\right)u_k^{\mathtt{l}}\right)^{n\Omega_{\mathtt{l}k}},u_k^{E_k}\right). \quad (39)$$

The factor 2^k-1 in the coef term takes into account that there are 2^k-1 non-zero symbols in a subspace of dimension k. We have to compute $\mathrm{P}\left(w\text{ is a codeword}\right)$, where $w\in W\left(E_1,\ldots,E_m\right)$ is some fixed word. Then

$$\mathrm{P}\left(w\text{ is a codeword}\right)=\frac{\left|\mathcal{G}\left(w\right)\right|}{\left|\mathrm{RESEGL}\left(n,\Omega,\Phi,m\right)\right|}, \quad (40)$$

where $\mathcal{G}\left(w\right)\subseteq\mathrm{RESEGL}\left(n,\Omega,\Phi,m\right)$ is the set of codes for which w is codeword. The total number of graphs in $\mathrm{RESEGL}\left(n,\Omega,\Phi,m\right)$ is given by

$$\left|\mathrm{RESEGL}\left(n,\Omega,\Phi,m\right)\right|=\prod_{k=0}^{m}\left(n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}k}\right)!g(m,k)^{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}k}}. \quad (41)$$

The factorial terms correspond to permutations of edges coming from variable nodes of different dimensions. There are $g(m,k)$ mappings which map a given subspace of dimension k to another given subspace of dimension k. This explains the power term.

Next, we count the number of graphs for which w is a codeword. This number is given by

$$|\mathcal{G}(w)| = \text{coef}\left(\prod_{\mathtt{r}}\prod_{s\in\mathcal{S}_{\mathtt{r}}} q_s(v_1,\ldots,v_m)^{n(1-r)\Phi_{\mathtt{r}s}}, \prod_{k=1}^{m} v_k^{E_k}\right) \left(n\sum \mathtt{l}\Omega_{\mathtt{l}0}\right)! \left(g(m,0)^{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}0}}\right) \prod_{k=1}^{m}\left(E_k!\left(n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}k}-E_k\right)!g(m,k)^{n\sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}k}-E_k}\right). \quad (42)$$

The factorial terms in (42) correspond to permuting edges among their class. This means that the edges which are attached to a variable node restricted to a subspace of dimension k and carrying a non-zero values can only be permuted among themselves. The power terms correspond to assigning permissible elements of GL_2^m to the edges which carry the value zero. The generating function $q_s(v_1,\ldots,v_m)$ for a check node with the state s is self explanatory by its definition. By combining (38, 39, 40, 41, 42), we get the desired result. ■

APPENDIX C
PROOF OF LEMMA 4.4

Proof: First consider the binary case. We show that for every residual graph with the desired degree distribution, the probability of erasure patterns resulting the graph is the same. Note that the edge connection in the original graph is same as in the residual graph. Consider the residual graph A. Take two variable nodes $\mathtt{v}_1$ and $\mathtt{v}_2$ and two edges e_1 (connected to $\mathtt{v}_1$) and e_2 (connected to $\mathtt{v}_2$). We swap the end of these edges on the check node side. Call this new graph A'. Consider the case when $\mathtt{v}_1$ and $\mathtt{v}_2$ are erased in A (their state is equal to 1). Clearly, the erasure pattern which results in A also results in A'. So, this is trivial. Consider the case when $\mathtt{v}_1$ and $\mathtt{v}_2$ are known in A. If $\mathtt{v}_1$ and $\mathtt{v}_2$ were also initially known then this erasure pattern also results in A' (by initially we mean the output of the channel). If the erasure pattern is such that $\mathtt{v}_1$ is known but $\mathtt{v}_2$ is erased, then for this erasure pattern we will construct another erasure pattern of equal probability for A'. We do this by making both $\mathtt{v}_1$ and $\mathtt{v}_2$ known initially, but we erase another variable node which was known in the erasure pattern of A. As $\mathtt{v}_2$ is known in A and was erased initially, it was revealed at some stage of the peeling decoder. Before this stage, the peeling decoder is identical for both A and A'. When $\mathtt{v}_2$ is revealed, all the edges connected to the check node are known. By backtracking the peeling decoder, we can find a node which was known initially. Now we erase this variable node and still the output of the peeling decoder is the same as in the case when $\mathtt{v}_2$ is known. If initially $\mathtt{v}_1$ and $\mathtt{v}_2$ are erased then we construct an erasure pattern where both $\mathtt{v}_1$ and $\mathtt{v}_2$ are known, but we erase two other variable nodes which were known in A. By similar arguments as in the previous case, it can again be shown that the peeling decoder will result in A'.

We apply the same sequence of arguments to the the non-binary case to prove the lemma. ■

APPENDIX D
PROOF OF LEMMA 5.3

Proof: To prove the lemma, we will consider the following cases

1) $u_1 \in [1,\infty)$ and $u_2 \in [0,1]$,
2) $u_2 \in [1,\infty)$ and either $u_1 \in [0,1]$ or $u_1 \in [1,\infty)$.

We will use the following facts and notations:

- $t_{-2}(u_2)(t_{-1}(u_1))$ is non-negative for $u_2(u_1) \in [0,1]$ and non-positive for $u_2(u_1) \in [1,\infty)$;

$t_{+2}(u_2)(t_{+1}(u_1))$ is non-negative for $\forall u_2(u_1) \in [0,\infty)$.

- By counting different types of edges, we get

$$(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}}\Phi_{\mathtt{r}s}n_1 = \sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}1}, \quad (43)$$

$$(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}}\Phi_{\mathtt{r}s}n_2 = \sum_{\mathtt{l}}\mathtt{l}\Omega_{\mathtt{l}2}. \quad (44)$$

1) Case $u_1 \in [1,\infty)$ and $u_2 \in [0,1]$. We have that $t_{-2}(u_2)$ is non-negative for $u_2 \in [0,1]$. Moreover,

$$u_1 t_{+1}(u_1) = t_{+1}\left(\frac{1}{u_1}\right), \quad t_{-1}\left(\frac{1}{u_1}\right) = -u_1 t_{-1}(u_1).$$

We use these relationships in $p\left(\frac{1}{u_1}\right)$ and obtain

$$p\left(\frac{1}{u_1}\right) = \frac{u_1^{n_1}}{4}(t_{+1}(u_1)^{n_1} + \sum_{i=1}^{3} t_{+1}(u_1)^{n_1(i)}(-t_{-1}(u_1))^{n_1-n_1(i)}).$$

As $t_{-1}(u_1)$ is non-positive for $u_1 \in [1,\infty)$, we get

$$p\left(\frac{1}{u_1}\right) \geq u_1^{n_1} p(u_1).$$

Rewriting $h_s(u_1,u_2)$ as

$$h_s(u_1,u_2) = p(u_1)t_{-2}(u_2)^{n_2} + \frac{t_{+1}(u_1)^{n_1}}{4}(t_{+2}(u_2)^{n_2} - t_{-2}(u_2)^{n_2}),$$

and using above relationships, we obtain

$$h_s\left(\frac{1}{u_1},u_2\right) \geq u_1^{n_1} h_s(u_1,u_2).$$

Substituting this and (43) in $\theta\left(\frac{1}{u_1},u_2\right)$, we obtain

$$\theta\left(\frac{1}{u_1},u_2\right) \geq \theta(u_1,u_2),$$

for $u_1 \in [1,\infty)$ and $u_2 \in [0,1]$.

2) Consider the case when $u_2 \in [1,\infty)$. To show that $\theta(u_1,u_2)$ does not attain its maximum when $u_2 \geq 1$ and $u_1 \in [0,\infty]$, we consider the sign of its first derivative over u_2. If it is non-positive for $u_2 \geq 1$, than the function in this region is decreasing and this will prove the desired result. Notice that here we consider the case when $n_2 > 0$ (when $n_2 = 0$, it is trivial to show that $\theta(u_1,u_2)$ is a decreasing function for $u_2 \geq 1$). Denote

$$a(u_1) = \frac{t_{+1}(u_1)^{n_1}}{4},$$

$$b(u_1) = \frac{1}{4}\sum_{i=1}^{3} t_{+1}(u_1)^{n_1(i)} t_{-1}(u_1)^{n_1-n_1(i)}.$$

Then after some calculations, we obtain the following expression for $\frac{\partial\theta}{\partial u_2}$:

$$\frac{\partial\theta}{\partial u_2} = \frac{1}{\ln 2}(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}}\frac{\Phi_{\mathtt{r}s}n_2}{t_{+2}(1)}C_1\cdot C_2, \quad (45)$$

where

$$C_1 = \sum_{l,j} l j \Omega_{l2}\Omega_{j2}\frac{u_2^{l-2}(3u_2^l - 3lu_2^j - l + 1)}{(1+3u_2^l)^2(1+3u_2^j)},$$

$$C_2 = \frac{3a(u_1)(u_2-1)t_{+2}(u_2)^{n_2-1}}{a(u_1)t_{+2}(u_2)^{n_2} + b(u_1)t_{-2}(u_2)^{n_2}} + \frac{b(u_1)(1+3u_2)t_{-2}(u_2)^{n_2-1}}{a(u_1)t_{+2}(u_2)^{n_2} + b(u_1)t_{-2}(u_2)^{n_2}}.$$

Consider the term C_1. We rewrite it as

$$C_1 = -\sum_{l}(l\Omega_{l2})^2\frac{(l-1)u_2^{l-2}}{(1+3u_2^l)^2} + \sum_{l>1}\sum_{j<l} l j \Omega_{l2}\Omega_{j2}\frac{C_3}{(1+3u_2^l)^2(1+3u_2^j)}$$

with

$$C_3 = -3u_2^{l+j-2}(3(j-1)u_2^l + 3(l-1)u_2^j + (l+j)) + 3(u_2^{j_2-2} + u_2^{l-2}) - (l-1)u_2^{l-2} - (j-1)u_2^{j-2}.$$

Note that the first sum gives us a negative term. The sign of the second term depends on the sign of C_3. The dominant terms in C_3 are u_2^{2j+l-2} and u_2^{2l+j-2} (remember that $u_2 \geq 1$) which have negative coefficients as $l > 1$. So, C_3 is non-positive for $u_2 \geq 1$, as well as the whole term C_1.

Now consider the term C_2. Notice that

$$\frac{b(u_1)}{a(u_1)} = \sum_{i=1}^{3}\left(\frac{t_{-1}(u_1)}{t_{+1}(u_1)}\right)^{n_1-n_1(i)}$$

while $0 \le \frac{t_{-1}(u_1)}{t_{+1}(u_1)} \le 1$ for $0 \le u_1 \le 1$ and $-1 \le \frac{t_{-1}(u_1)}{t_{+1}(u_1)} < 0$ for $u_1 > 1$. Therefore, $0 \le \frac{b(u_1)}{a(u_1)} \le 3$ for $u_1 \in [0,1]$ and $-3 \le \frac{b(u_1)}{a(u_1)} < 0$ for $u_1 > 1$. Then we have

$$a(u_1)t_{+2}(u_2)^{n_2} + b(u_1)t_{-2}(u_2)^{n_2} \ge a(u_1)(t_{+2}(u_2) + 3t_{-2}(u_2))$$
$$= \sum_l l\Omega_{l2} \frac{4}{1+3u_2^l} \ge 0.$$

for $u_2 \ge 1$, thus the denominator of C_2 is positive.

We need to consider the numerator of C_2. We will treat two cases $n_2 > 2$ and $n_2 \le 2$ separately. Let $n_2 > 2$. Notice that when $b(u_1)t_{-2}(u_2)^{n_2-1}$ is positive, then the numerator is positive as well. We only need to verify the cases when $b(u_1)t_{-2}(u_2)^{n_2-1}$ is negative, i.e. either when $u_1 \le 1$ and n_2 is even, either when $u_1 \ge 1$ and n_2 is odd. The expression under consideration takes its smallest values for $n_2 = 3$ and $u_1 \ge 1$:

$$a(u_1)t_{+2}(u_2)^{n_2-1}3[u_2-1] + b(u_1)t_{-2}(u_2)^{n_2-1}[1+3u_2]$$
$$\ge a(u_1)[t_{+2}^2(u_2)(3u_2-3) - 3t_{-2}^2(u_2)(1+3u_2)]$$
$$= 6a(u_1)\sum_{l,j} l\,j\Omega_{l2}\Omega_{j2}[\frac{3u_2^{l+j-1} - 5u_2^{l+j-2} + 3u_2^l + 3u_2^j}{(1+3u_2^l)(1+3u_2^j)}$$
$$- \frac{u_2^{l-1} + u_2^{j-1} + u_2 + 1}{(1+3u_2^l)(1+3u_2^j)}],$$

which is non-negative for $u_2 \ge 1$ as the polynomial $3u_2^{l+j-1} - 5u_2^{l+j-2} + 3u_2^l + 3u_2^j - u_2^{l-1} - u_2^{j-1} - u_2 - 1$ is non-negative. Thus, in the case of $n_2 > 2$ the term C_2 in (45) is non-negative which implies that $\theta(u_1,u_2)$ is a decreasing function when $u_2 \ge 1$.

Now let us consider a special case when $n_2 \le 2$. Then the sign of $\frac{\partial\theta}{\partial u_2}$ can be both positive and negative depending on u_1 and u_2, and our previous reasoning works only if $u_1 \in [0,1)$ (the numerator of C_2 is positive both for $n_2 = 1$ and $n_2 = 2$). So suppose $u_1 \ge 1$.

Using (43, 44) we rewrite the expression for $\theta(u_1,u_2)$

$$\theta(u_1,u_2) =$$
$$(1-R)\sum_{\mathtt{r}}\sum_{s\in\mathcal{S}_{\mathtt{r}}} \Phi_{\mathtt{r}s} \log\left\{ \frac{2^{\dim(s)}}{4}\left[g_{+1}(u_1)^{n_1} g_{+2}(u_2)^{n_2}\right.\right.$$
$$\left.\left. + \; g_{-2}(u_2)^{n_2}\left(\frac{4p(u_1)f_1(u_1)^{n_1}}{t_{+1}(1)^{n_1}} - g_{+1}(u_1)^{n_1}\right)\right]\right\}, \quad (46)$$

where the functions $g_{+1}(u_1), g_{-1}(u_1), g_{+2}(u_2)$ and $g_{-2}(u_2)$ are defined as

$$g_{\pm1}(u_1) = f_1(u_1)\frac{t_{\pm1}(u_1)}{t_{+1}(1)}, \quad g_{\pm2}(u_2) = f_2(u_2)\frac{t_{\pm2}(u_2)}{t_{+2}(1)},$$

with

$$f_1(u_1) = \prod_{\mathtt{l}}\left(\frac{1+u_1^{\mathtt{l}}}{2}\right)^{\frac{\Omega_{\mathtt{l}1}}{t_{+1}(1)}}, \quad f_2(u_2) = \prod_{\mathtt{l}}\left(\frac{1+3u_2^{\mathtt{l}}}{4}\right)^{\frac{\Omega_{\mathtt{l}2}}{t_{+2}(1)}}.$$

The g functions satisfy the following properties.

$$0 \le g_{+1(u_1)} \le 1, \forall u_1 \in [1,\infty) \quad g_{-1}(u_1) \le g_{+1}(u_1), \forall u_1 \in [1,\infty),$$
$$0 \le g_{+2}(u_2) \le 1, \forall u_2 \in [1,\infty).$$

First let $n_2 = 1$. The first condition implies dim(s)=2. Also, at least two different subspaces of dimension 1 are present. Let $z(u_1,u_2)$ be the expression inside the logarithm in (46). Then

$$z(u_1,u_2) = \frac{f_2(u_2)}{t_{+2}(1)}\sum_{\mathtt{l}}\frac{\mathtt{l}\Omega_{\mathtt{l}2}}{1+3u_2^{\mathtt{l}}}$$
$$\left\{\sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)} + g_{+1}(u_1)^{n_1} + \right.$$
$$\left. u_2^{\mathtt{l}-1}\left(3g_{+1}(u_1)^{n_1} - \sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}\right)\right\}. \quad (47)$$

If we show that $z(u_1,u_2) \le g_{+2}(u_2)$, then we prove that $z(u_1,u_2) \le 1$ as $g_{+2}(u_2) \le 1$. By (47) it suffices to prove

$$\sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)} + g_{+1}(u_1)^{n_1} + u_2^{1-1}\left(3g_{+1}(u_1)^{n_1} - \sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}.\right) \le 1 + 3u_2^{1-1},$$

which simplifies to

$$\sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)} + g_{+1}(u_1)^{n_1} - 1 \le u_2^{1-1}\left(3 - 3g_{+1}(u_1)^{n_1} + \sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}\right).$$

As $u_2 \ge 1$ and the coefficient of u_2^{1-1} is positive, the above equation holds if

$$\sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)} + g_{+1}(u_1)^{n_1} - 1 \le 3 - 3g_{+1}(u_1)^{n_1} + \sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}, \tag{48}$$

which simplifies to

$$g_{+1}(u_1)^{n_1} - 1 \le 3 - 3g_{+1}(u_1)^{n_1}.$$

This holds as $g_{+1}(u_1) \le 1$ for $u_1 \in [1,\infty)$, so we proved the desired result for $n_2 = 1$.

Now let $n_2 = 2$. The expression for $z(u_1,u_2)$ becomes

$$z(u_1,u_2) = \frac{f_2(u_2)^2}{t_{+2}(1)^2} \sum_{1,j} \frac{1 j \Omega_{12}\Omega_{j2}}{(1+3u_2^1)(1+3u_2^j)} s(u_1,u_2)$$

with

$$s(u_1,u_2) = g_{+1}(u_1)^{n_1}[2u_2^{l-1} + 2u_2^{j-1} + 8u_2^{l+j-2}] + \sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}[1 - u_2^{l-1} - u_2^{j-1} + u_2^{l+j-2}].$$

Then, if we show that $s(u_1,u_2) \le (1+3u_2^{l-1})(1+3u_2^{j-1})$, then $z(u_1,u_2) \le 1$ and we prove the desired result. As

$$-3 \le \frac{\sum_{i=1}^{3} g_{+1}(u_1)^{n_1(i)} g_{-1}(u_1)^{n_1-n_1(i)}}{g_{+1}(u_1)^{n_1}} \le 0$$

for $u_1 \ge 1$, we have

$$\begin{aligned} s(u_1,u_2) &\le g_{+1}(u_1)^{n_1}[2u_2^{l-1} + 2u_2^{j-1} + 8u_2^{l+j-2}] \\ &\le 2u_2^{l-1} + 2u_2^{j-1} + 8u_2^{l+j-2} \\ &\le (1+3u_2^{l-1})(1+3u_2^{j-1}) \end{aligned}$$

as needed.

Now we need to prove (23). Consider $|\theta - \tilde{\theta}|$ where θ and $\tilde{\theta}$ are expressed as it is shown in (46), i.e. in the form

$$\theta = (1-R)\sum_{\mathtt{r}} \sum_{s \in \mathcal{S}_{\mathtt{r}}} \Phi_{\mathtt{r}s} \log_2 z(\Omega,u_1,u_2),$$

where we denote $z(u_1,u_2)$ by $z(\Omega,u_1,u_2)$, to stress the dependence on Ω. By using

$$|a_1a_2 - b_1b_2| \le |a_1 - b_1||a_2 + b_2| + |a_1 + b_1||a_2 - b_2|,$$

we obtain that

$$|\theta - \tilde{\theta}| \le (1-R)\sum_{\mathtt{r}} \sum_{s \in \mathcal{S}_{\mathtt{r}}} |\Phi_{\mathtt{r}s} - \tilde{\Phi}_{\mathtt{r}s}||\log_2 z(\Omega,u_1,u_2) + \log_2 z(\tilde{\Omega},u_1,u_2)| + |\Phi_{\mathtt{r}s} + \tilde{\Phi}_{\mathtt{r}s}||\log_2 z(\Omega,u_1,u_2) - \log_2 z(\tilde{\Omega},u_1,u_2)|.$$

Consider the term

$$|\log_2 z(\Omega,u_1,u_2) + \log_2 z(\tilde{\Omega},u_1,u_2)|.$$

Let also $n_2 = 0$ and $n_1 > 0$, which yields

$$z(\Omega,u_1,u_2) = z(\Omega_1,u_1,u_2) = g_{+1}(u_1)^{n_1} + g_{-1}(u_1)^{n_1}.$$

First, let us show that

$$(\log z(\Omega_1,u_1,u_2))^2 \le c^2(1-u_1)^4, \tag{49}$$

where c is a positive constant. We consider the sign of the derivative

$$\frac{\partial}{\partial u_1}\{c^2(1-u_1)^4-(\log z)^2\} = -4c^2(1-u_1)^3 - 2\frac{\log z}{z}\frac{\partial z}{\partial u_1}$$

Notice that $\log z|_{u_1=1} = 0$. Moreover, for $n_1 \geq 2$,

$$\frac{\partial \log z}{\partial u_1}|_{u_1=0} = 0, \quad \frac{\partial^2 \log z}{\partial u_1^2}|_{u_1=0} = 0.$$

If we expand $z(\Omega_1, u_1, u_2)$ at $u_1 = 1$, the biggest term of the Taylor series in u_1 is of the order $(u_1-1)^3$. Hence, by choosing the constant c large enough, we obtain

$$\frac{\partial}{\partial u_1}\{c^2(1-u_1)^4-(\log z)^2\} \leq 0$$

, for $u_1 \in [0,1]$. As the function $c^2(1-u_1)^4-(\log z)^2$ is 0 for $u_1 = 1$, so is positive for $u_1 \in [0,1)$ and that (49) holds. This implies

$$|\log_2 z(\Omega_1, u_1, u_2) + \log_2 z(\tilde{\Omega}_1, u_1, u_2)| \leq c(1-u_1)^2,$$

as desired. By using the same argument, we can show that the sum of two logarithms is $\leq c(1-u_2)^2$ when $n_1 = 0$ and $n_2 > 0$ and that it is $\leq c(1-u_1)(1-u_2)$ when $n_1 > 0$ and $n_2 > 0$.

Now consider the term

$$|\log_2 z(\Omega, u_1, u_2) - \log_2 z(\tilde{\Omega}, u_1, u_2)|.$$

Once again, let $n_2 = 0$ and $n_1 > 0$ first. Denote by

$$T = \frac{\log_2 z(\Omega, u_1, u_2)}{(1-u_1)^2}.$$

By using the multivariate mean value theorem (p.172, [24]) with respect to Ω's, we obtain that

$$|T - \tilde{T}| \leq d(\{\Omega, \Phi\}, \{\tilde{\Omega}, \Phi\}) \max_{\Omega_1', \Omega_2'} |\nabla T(\Omega_1', \Omega_2')|$$

with

$$\nabla T = \left(\frac{\partial T}{\partial \Omega_{11}}, \ldots, \frac{\partial T}{\partial \Omega_{l_{max}1}}, \frac{\partial T}{\partial \Omega_{12}}, \ldots, \frac{\partial T}{\partial \Omega_{l_{max}2}} \right).$$

One can be shown that $\frac{\partial T}{\partial \Omega_i}$ is a continuous function for $u_1 \in [0,1]$ for any $\Omega_i \in \Omega$, therefore, $\max_{\Omega_1', \Omega_2'} |\nabla T(\Omega_1', \Omega_2')|$ can be bounded by some constant $c_1 > 0$. This implies

$$|\log_2 z(\Omega, u_1, u_2) - \log_2 z(\tilde{\Omega}, u_1, u_2)| \leq c_1 d(\{\Omega, \Phi\}, \{\tilde{\Omega}, \Phi\})(1-u_1)^2.$$

The bounds $\leq cd(\{\Omega, \Phi\}, \{\tilde{\Omega}, \Phi\})(1-u_2)^2$ when $n_1 = 0$ and $n_2 > 0$ and $\leq cd(\{\Omega, \Phi\}, \{\tilde{\Omega}, \Phi\})(1-u_1)(1-u_2)$ when $n_1 > 0$ and $n_2 > 0$ can be obtained in the same way.

■