Design of LDPC Codes: A Survey and New Results

Gianluigi Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, and William E. Ryan

Original scientific paper

***Abstract*— This survey paper provides fundamentals in the design of LDPC codes. To provide a target for the code designer, we first summarize the EXIT chart technique for determining (near-)optimal degree distributions for LDPC code ensembles. We also demonstrate the simplicity of representing codes by protographs and how this naturally leads to quasi-cyclic LDPC codes. The EXIT chart technique is then extended to the special case of protograph-based LDPC codes. Next, we present several design approaches for LDPC codes which incorporate one or more accumulators, including quasi-cyclic accumulator-based codes. The second half the paper then surveys several algebraic LDPC code design techniques. First, codes based on finite geometries are discussed and then codes whose designs are based on Reed-Solomon codes are covered. The algebraic designs lead to cyclic, quasi-cyclic, and structured codes. The masking technique for converting regular quasi-cyclic LDPC codes to irregular codes is also presented. Some of these results and codes have not been presented elsewhere. The paper focuses on the binary-input AWGN channel (BI-AWGNC). However, as discussed in the paper, good BI-AWGNC codes tend to be universally good across many channels. Alternatively, the reader may treat this paper as a starting point for extensions to more advanced channels. The paper concludes with a brief discussion of open problems.**

I. Introduction

The class of low-density parity-check (LDPC) codes represents the leading edge in modern channel coding. They have held the attention of coding theorists and practitioners in the past decade because of their near-capacity performance on a large variety of data transmission and storage channels and because their decoders can be implemented with manageable complexity. They were invented by Gallager in his 1960 doctoral dissertation [1] and were scarcely considered in the 35 years that followed. One notable exception is Tanner, who wrote an important paper in 1981 [2] which generalized LDPC codes and introduced a graphical representation of LDPC codes, now called Tanner graphs. Apparently independent of Gallager's work, LDPC codes were re-invented in the mid-1990's by MacKay, Luby, and others [3][4][5][6] who noticed the advantages of linear block codes which possess sparse (low-density) parity-check matrices.

This papers surveys the state-of-the-art in LDPC code design for binary-input channels while including a few new results as well. While it is tutorial in some aspects, it is not entirely a tutorial paper, and the reader is expected to be fairly versed on the topic of LDPC codes. Tutorial coverages of LDPC codes can be found in [7][8]. The purpose of this paper is to give the reader a detailed overview of various LDPC code design approaches and also to point the reader to the literature. While our emphasis is on code design for the binary-input AWGN channel (BI-AWGNC), the results in [9][10][11][12] demonstrate that a LDPC code that is good on the BI-AWGNC tends to be universally good and can be expected to be good on most wireless, optical, and storage channels.

We favor code designs which are most appropriate for applications, by which we mean codes which have low-complexity encoding, good waterfall regions, and low error floors. Thus, we discuss quasi-cyclic (QC) codes because their encoders may be implemented by shift-register circuits [13]. We also discuss accumulator-based codes because low-complexity encoding is possible from their parity-check matrices, whether they are quasi-cyclic or not. The code classes discussed tend to be the ones (or related to the ones) used in applications or adopted for standards. Due to time and space limitations, we cannot provide a complete survey. The present survey is biased toward the expertise and interests of the authors.

Before a code can be designed, the code designer needs to know the design target. For this reason, Section II first briefly reviews the belief propagation decoder for LDPC codes and then presents the so-called extrinsic information transfer (EXIT) chart technique for this decoder. The EXIT chart technique allows one to obtain near-optimal parameters for LDPC code ensembles which guide the code designer. The EXIT technique is extended in Section III to the case of codes based on protographs. Section IV considers LDPC codes based on accumulators. The code types treated in that section are: repeat-accumulate, irregular repeat-accumulate, irregular repeat-accumulate-accumulate, generalized irregular repeat-accumulate, and accumulate-repeat-accumulate. That section also gives examples of quasi-cyclic code design using protograph (or base matrix) representations. Section V surveys the literature on cyclic and quasi-cyclic LDPC code design based on finite geometries. Section VI presents several LDPC code design techniques based on Reed-Solomon codes. Section VII presents the masking technique for converting regular QC codes to irregular QC codes to conform to prescribed code parameters. Section VIII contains some concluding remarks and some open problems.

II. Design via EXIT Charts

We start with an $m \times n$ low-density parity-check matrix $\mathbf{H}$, which corresponds to a code with design rate $(n-m)/n$,

Manuscript received July 04, 2006; revised August 25, 2006. This work was supported by the University of Bologna, NASA-Goddard, and NSF.

This paper has been approved by F. Chiaraluce.

Gianluigi Liva is with the University of Bologna (email: gliva@deis.unibo.it).

Shumei Song, Lan Lan, and Shu Lin are with the University of California at Davis (e-mail: ssmsong@ece.ucdavis.edu, squashlan@gmail.com, shulin@ece.ucdavis.edu).

Yifei Zhang and William E. Ryan are with the University of Arizona, U.S.A. (e-mail: {yifeiz, ryan}@ece.arizona.edu).

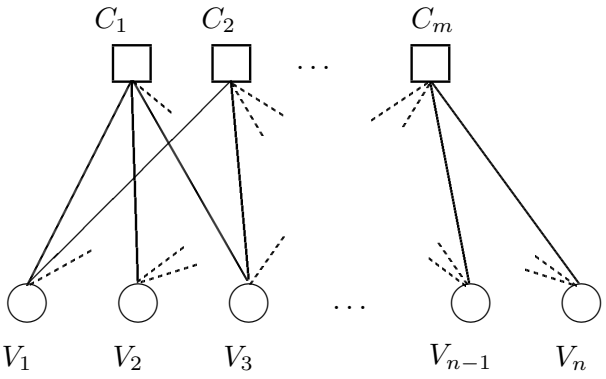

Fig. 1. Tanner graph representation of LDPC codes.

which could be less than the actual rate, $R = k/n$, where k is the number of information bits per codeword. $\mathbf{H}$ gives rise to a Tanner graph which has m check nodes, one for each row of $\mathbf{H}$, and n variable nodes, one for each column of $\mathbf{H}$. Considering the general case in which $\mathbf{H}$ has non-uniform row and column weight, the Tanner graph can be characterized by degree assignments $\{d_v(i)\}_{i=1}^{n}$ and $\{d_c(j)\}_{j=1}^{m}$, where $d_v(i)$ is the degree of the i-th variable node and $d_c(j)$ is the degree of the j-th check node. Such a graph, depicted in Fig. 1, is representative of the iterative decoder, with each node representing a soft-in/soft-out processor (or node decoder).

We shall assume the BI-AWGNC in our description of the LDPC iterative decoder. In this model, a received channel sample y is given by $y = x + w$, where $x = (-1)^c \in \{\pm 1\}$ is the bipolar representation of the transmitted code bit $c \in \{0, 1\}$ and w is a white Gaussian noise sample distributed as $\eta\left(0, \sigma_w^2\right)$, where $\sigma_w^2 = N_0/2$, following convention. The channel bit log-likelihood ratios (LLRs) are computed as

$$L_{ch} = \log\left(\frac{p\left(x = +1 \mid y\right)}{p\left(x = -1 \mid y\right)}\right) = \frac{2y}{\sigma_w^2}. \tag{1}$$

In one iteration of the conventional, flooding-schedule iterative decoder, the variable node decoders (VNDs) first process their input LLRs and send the computed outputs (messages) to each of their neighboring check node decoders (CNDs); then the CNDs process their input LLRs and send the computed outputs (messages) to each of their neighboring VNDs. More specifically, the message from the i-th VND to the j-th CND is

$$L_{i\to j} = L_{ch,i} + \sum_{j' \neq j} L_{j'\to i} \tag{2}$$

where $L_{j'\to i}$ is the incoming message from CND j' to VND i and where the summation is over the $d_v(i) - 1$ check node neighbors of variable node i, excluding check node j. The message from CND j to VND i is given by

$$L_{j\to i} = 2\tanh^{-1}\left(\prod_{i' \neq i} \tanh\left(L_{i'\to j}\right)\right) \tag{3}$$

where $L_{i'\to j}$ is the incoming message from VND i' to CND j and where the product is over the $d_c(j) - 1$ variable node neighbors of check node j, excluding variable node i . This

We now discuss the EXIT chart technique [14][15][11] for this decoder and channel model. The idea is that the VNDs and the CNDs work cooperatively and iteratively to make bit decisions, with the metric of interest generally improving with each half-iteration. A transfer curve which plots the input metric versus the output metric can be obtained for both the VNDs and the CNDs, where the transfer curve for the VNDs depends on the channel SNR. Further, since the output metric for one processor is the input metric for its companion processor, one can plot both transfer curves on the same axes, but with the abscissa and ordinate reversed for one processor. Such a chart aids in the prediction of the *decoding threshold* of the ensemble of codes characterized by given VN and CN degree distributions: the decoding threshold is the SNR at which the two transfer curves just touch, precluding convergence of the two processors. EXIT chart computations are thus integral to the optimization of Tanner graph node degree distributions for LDPC codes and are the main computation in the optimization process. We emphasize that decoding threshold prediction techniques such as EXIT charts or density evolution [16] assume a graph with no cycles, an infinite codeword length, and an infinite number of decoding iterations.

An EXIT chart example is depicted in Fig. 2 for the ensemble of regular LDPC codes on the BI-AWGNC with $d_v(i) = d_v = 3$ for $i = 1, ..., n$, and $d_c(j) = d_c = 6$ for $j = 1, ..., m$. In the figure, the metric used for the transfer curves is extrinsic mutual information, giving rise to the name extrinsic information transfer (EXIT) chart. (The notation used in the figure is explained below.) Also shown in the figure is the decoding trajectory corresponding to these EXIT curves. As the SNR increases, the top curve shifts upwards, increasing the "tunnel" between the two curves and thus the decoder convergence rate. The SNR for this figure is just above the decoding threshold for codes with $(d_v, d_c) = (3, 6)$, $(E_b/N_0)_{thres} = 1.1$ dB. Other metrics, such as SNR and mean [17][18] and error probability [19] are possible, but mutual information generally gives the most accurate prediction of the decoding threshold [14][20] and is a universally good metric across many channels [9][10][11][12].

To facilitate EXIT chart computations, the following Gaussian assumption is made. First, we note that the LLR L_{ch} in (1) corresponding to the BI-AWGNC is Gaussian with mean $\mu_{ch} = 2x/\sigma_w^2$ and variance $\sigma_{ch}^2 = 4/\sigma_w^2$. From this and the usual assumption that the all-zeros codeword was transmitted (thus, $x_i = +1$ for $i = 1, ..., n$), $\sigma_{ch}^2 = 2\mu_{ch}$. This is equivalent to the *symmetric condition* of [16] which states that the conditional pdf of an LLR value L must satisfy $p_L\left(l \mid x\right) = p_L\left(-l \mid x\right) e^{xl}$. Now, it has been observed that under normal operating conditions and after a few iterations, the LLRs $L_{i\to j}$ and $L_{j\to i}$ are approximately Gaussian and, further, if they are assumed to be symmetric-Gaussian, as is the case for L_{ch}, the decoding threshold predictions are very accurate (e.g., when compared to the more accurate, but more computationally intensive density evolution results [16]). Moreover, the symmetric-Gaussian assumption vastly simplifies EXIT chart analyses.

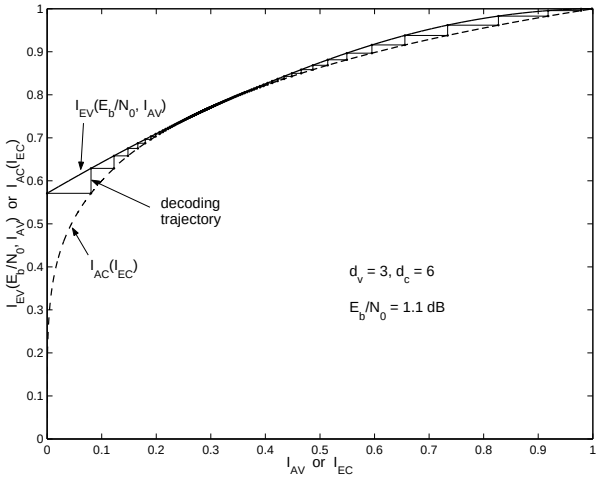

Fig. 2. EXIT chart example for $(d_v, d_c) = (3, 6)$ regular LDPC code.

for both VNDs and the CNDs, first for regular LDPC codes and then for irregular codes. Following [14][15], excluding the inputs from the channel, we consider VND and CND inputs to be *a priori* information, designated by 'A', and their outputs to be extrinsic information, designated by 'E'. Thus, an extrinsic information transfer curve for the VNDs plots the extrinsic information I_E as a function of its input *a priori* information, I_A, and similarly for the CNDs.

The VND EXIT curve, $I_{E,V}$ versus $I_{A,V}$, under the symmetric-Gaussian assumption for VND inputs, $L_{ch,i}$ and $\{L_{j'\to i}\}$, and outputs, $L_{i\to j}$, can be obtained as follows. From (2) and an independent-message assumption, $L_{i\to j}$ is Gaussian with variance $\sigma^2 = \sigma_{ch}^2 + (d_v - 1)\,\sigma_A^2$ (hence, mean $\sigma^2/2$). The mutual information between the random variable X (corresponding to the realization x_i) and the extrinsic LLR $L_{i\to j}$ is therefore (for simplicity, we write L for $L_{i\to j}$, x for x_i, and $p_L(l \mid \pm)$ for $p_L(l \mid x = \pm 1)$)

$$\begin{aligned} I_{E,V} &= H(X) - H(X \mid L) \\ &= 1 - E\left[\log_2\left(1/p_{X\mid L}(x \mid l)\right)\right] \\ &= 1 - \sum_{x=\pm 1} \frac{1}{2} \int_{-\infty}^{\infty} p_L(l \mid x) \\ &\quad \cdot \log_2\left(\frac{p_L(l \mid +) + p_L(l \mid -)}{p_L(l \mid x)}\right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l \mid +) \log\left(1 + \frac{p_L(l \mid -)}{p_L(l \mid +)}\right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l \mid +) \log\left(1 + e^{-l}\right) dl \end{aligned}$$

where the last line follows from the symmetry condition and because $p_L(l \mid x = -1) = p_L(-l \mid x = +1)$ for Gaussian densities.

Since $L_{i\to j} \sim \eta\left(\sigma^2/2, \sigma^2\right)$ (when conditioned on $x_i = +1$), we have

$$I_{E,V} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(l - \sigma^2/2\right)^2/2\sigma^2} \log\left(1 + e^{-l}\right) dl \ .$$

For convenience we write this as

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)\,\sigma_A^2 + \sigma_{ch}^2}\right) \ , \qquad (5)$$

following [15]. To plot $I_{E,V}$ versus $I_{A,V}$, where $I_{A,V}$ is the mutual information between the VND inputs $L_{j\to i}$ and the channel bits x_i, we apply the symmetric-Gaussian assumption to these inputs so that

$$I_{A,V} = J(\sigma_A) \qquad (6)$$

and

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)\left[J^{-1}(I_{A,V})\right]^2 + \sigma_{ch}^2}\right) \ . \qquad (7)$$

The inverse function $J^{-1}(\cdot)$ exists since $J(\sigma_A)$ is monotonic in σ_A. Lastly, $I_{E,V}$ can be parameterized by E_b/N_0 for a given code rate R since $\sigma_{ch}^2 = 4/\sigma_w^2 = 8R\,(E_b/N_0)$. Approximations of the functions $J(\cdot)$ and $J^{-1}(\cdot)$ are given in [15].

To obtain the CND EXIT curve, $I_{E,C}$ versus $I_{A,C}$, we can proceed as we did in the VND case, e.g., begin with the symmetric-Gaussian assumption. However, this assumption is not sufficient because determining the mean and variance for a CND output $L_{j\to i}$ is not straightforward, as is evident from the computation for CNDs in (3). Closed-form expressions have been derived for the check node EXIT curves [21][22]. Computer-based numerical techniques can also be used to obtain these curves. However, the simplest technique exploits the following duality relationship (proven to be exact for the binary erasure channel [11]): the EXIT curve for a degree-d_c check node (i.e., rate-$(d_c - 1)/d_c$ single-parity check (SPC) code) and that of a degree-d_c variable node (i.e., rate-$1/d_c$ repetition code) are related as

$$I_{E,SPC}(d_c, I_A) = 1 - I_{E,REP}(d_c, 1 - I_A) \, .$$

This relationship was shown to be very accurate for the BI-AWGNC in [21][22]. Thus,

$$\begin{aligned} I_{E,C} &= 1 - I_{E,V}(\sigma_{ch} = 0, d_v \leftarrow d_c, I_{A,V} \leftarrow 1 - I_{A,C}) \\ &= 1 - J\left(\sqrt{(d_c - 1)\left[J^{-1}(1 - I_{A,C})\right]^2}\right) . \end{aligned} \qquad (8)$$

For irregular LDPC codes, $I_{E,V}$ and $I_{E,C}$ are computed as weighted averages. The weighting is given by the coefficients of the "edge perspective" degree distribution polynomials $\lambda(z) = \sum_{d=1}^{d_v} \lambda_d z^{d-1}$ and $\rho(z) = \sum_{d=1}^{d_c} \rho_d z^{d-1}$, where λ_d is the fraction of edges in the Tanner graph connected to degree-d variable nodes, ρ_d is the fraction of edges connected to degree-d check nodes, and $\lambda(1) = \rho(1) = 1$. Then, for irregular LDPC codes,

$$I_{E,V} = \sum_{d=1}^{d_v} \lambda_d I_{E,V}(d, I_{A,V}) \qquad (9)$$

where $I_{E,V}(d)$ is given by (7) with d_v replaced by d, and

$$I_{E,C} = \sum_{d=1}^{d_c} \rho_d I_{E,C}(d, I_{A,C}) \qquad (10)$$

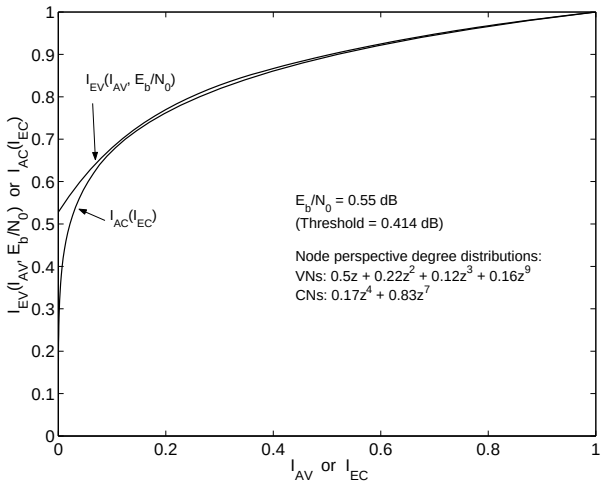

Fig. 3. EXIT chart for rate-1/2 irregular LDPC code. (Ack: S. AbuSurra)

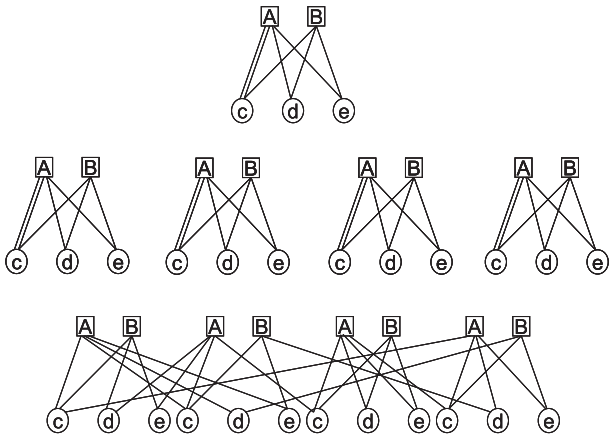

Fig. 4. Illustration of the protograph copy and permute procedure with $q = 4$ copies.

It has been shown [11] that to optimize the decoding threshold on the binary erasure channel, the shapes of the VND and CND transfer curves must be well matched in the sense that the CND curve fits inside the VND curve (an example will follow). This situation has also been observed on the BI-AWGNC [15]. Further, to achieve a good match, the number of different VN degrees need only be about 3 or 4 and the number of different CN degrees need only be 1 or 2.

Example 1: We consider the design of a rate-1/2 irregular LDPC code with four possible VN degrees and two possible CN degrees. Given than $\lambda(1) = \rho(1) = 1$ and $R = 1 - \int_0^1 \rho(z)dz / \int_0^1 \lambda(z)dz$ [16],[4], only two of the four coefficients for $\lambda(z)$ need be specified and only one of the two for $\rho(z)$ need be specified. A non-exhaustive search yielded $\lambda(z) = 0.267z + 0.176z^2 + 0.127z^3 + 0.430z^9$ and $\rho(z) = 0.113z^4 + 0.887z^7$ with a decoding threshold of $(E_b/N_0)_{thres} = 0.414$ dB. The EXIT chart for $E_b/N_0 = 0.55$ dB is presented in Fig. 3. The figure also gives the "node perspective" degree distribution information. □

The references contain additional information on EXIT charts, including the so-called area property, EXIT charts for the Rayleigh channel, for higher-order modulation, and for multi-input/multi-output channels [14][15][11][23].

III. Design of Protograph-Based Codes

A. Definition and Problem Statement

A protograph [24][25][26][27] is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied Q times, and then the edges of the individual replicas are permuted among the replicas (under restrictions described below) to obtain a single, large graph. An example is presented in Fig. 4. The permuted edge connections are specified by the parity-check matrix $\mathbf{H}$. Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labeled so that if variable node V is connected to check node C in the protograph, then variable node V in a replica can only preserves the decoding threshold properties of the protograph. A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. For LDPC codes, the copy-and-permute procedure must eliminate such parallel connections in order to obtain a derived graph appropriate for a parity-check matrix.

It is convenient to choose the parity-check matrix $\mathbf{H}$ as an $M \times N$ array of $Q \times Q$ (weight-one) circulant permutation matrices (some of which may be the $Q \times Q$ zero matrix). When $\mathbf{H}$ is an array of circulants, the LDPC code will be quasi-cyclic. Such a structure has a favorable impact on both the encoder and the decoder. The encoder for QC codes can be implemented with shift-register circuits with complexity linearly proportional to m for serial encoding and to n for parallel encoding [13]. By contrast, encoders for unstructured LDPC codes require much more work. The decoder for QC LDPC codes can be implemented in a modular fashion by exploiting the circulant-array structure of $\mathbf{H}$ [28][29].

Below we present an extension of the EXIT approach to codes defined by protographs. This extension is a multi-dimensional numerical technique and as such does not have a two-dimensional EXIT chart representation of the iterative decoding procedure. Still, the technique yields decoding thresholds for LDPC code ensembles specified by protographs. This multi-dimensional technique is facilitated by the relatively small size of protographs and permits the analysis of protograph code ensembles characterized by the presence of *critical node types*, i.e., node types which can lead to failed EXIT-based convergence of code ensembles. Examples of critical node types are degree-1 variable nodes and punctured variable nodes.

A code ensemble specified by a protograph is a refinement (sub-ensemble) of a code ensemble specified simply by the protograph's (hence, LDPC code's) degree distributions. To demonstrate this, we introduce the adjacency matrix $\mathbf{B} = [b_{ji}]$ for a protograph, also called a base matrix [25], where b_{ji} is the number of edges between CN j and VN i. As an example, for the protograph at the top of Fig. 4,

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 1 \end{pmatrix}$$

Consider also an alternative protograph and base matrix specified by

$$\mathbf{B}' = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix} .$$

The degree distributions of both of these protographs are identical and are easily seen to be

$$\begin{aligned} \lambda(z) &= \frac{4}{7}z + \frac{3}{7}z^2 \\ \rho(z) &= \frac{3}{7}z^2 + \frac{4}{7}z^3. \end{aligned}$$

However, the ensemble corresponding to $\mathbf{B}$ has a threshold of $E_b/N_0 = 0.78$ dB and that corresponding to $\mathbf{B}'$ has a threshold at 0.83 dB. (For reference, density evolution [16] applied to the above degree distributions gives 0.817 dB.)

As another example, let

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\mathbf{B}' = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} ,$$

noting that they have identical degree distributions. We also puncture the bits corresponding to the second column in each base matrix. Using the multidimensional EXIT algorithm described below, the thresholds for $\mathbf{B}$ and $\mathbf{B}'$ in this case were computed to be 0.48 dB and $+\infty$, respectively.

Thus, standard EXIT analysis based on degree distributions is inadequate for protograph-based LDPC code design. In fact, the presence of degree-1 variable nodes as in our second example implies that there is a term in the summation in (9) of the form

$$\lambda_1 I_{E,V}(1, I_{A,V}) = J(\sigma_{ch}) .$$

Since $J(\sigma_{ch})$ is always less than one for $0 < \sigma_{ch} < \infty$ and since $\sum_{d=1}^{d_v} \lambda_d = 1$, the summation in (9), that is, $I_{E,V}$, will be strictly less than one. Again, standard EXIT analysis implies failed convergence for codes with the same degree distributions as $\mathbf{B}$ and $\mathbf{B}'$. This is in contrast with the fact that codes in the $\mathbf{B}$ ensemble do converge when the SNR exceeds the threshold of 0.48 dB.

In the following, a multidimensional EXIT technique [30][31] will be presented which overcomes this issue and allows the determination of the decoding threshold for codes based on protographs (possibly with punctured nodes).

B. Multidimensional EXIT Analysis

The algorithm presented in [30][31] eliminates the average in (9) and considers the propagation of the messages on a decoding tree which is specified by the protograph of the ensemble. Let $\mathbf{B} = [b_{ji}]$ be the $M \times N$ base matrix for the protograph under analysis. Let $I_{E,V}^{i \to j}$ be the extrinsic mutual information between code bits associated with "type i" VNs and the LLRs $L_{i \to j}$ sent from these VNs to "type j" CNs. code bits associated with "type i" VNs and the LLRs $L_{j \to i}$ sent from "type j" CNs to these VNs. Then, because $I_{E,C}^{j \to i}$ acts as *a priori* mutual information in the calculation of $I_{E,V}^{i \to j}$, following (7) we have (given an edge exists between CN j and VN i, i.e., given $b_{ji} \neq 0$)

$$I_{E,V}^{i \to j} = J\left(\sqrt{\sum_{c=1}^{M} (b_{ci} - \delta_{cj}) \left(J^{-1}(I_{E,C}^{c \to i}) \right)^2 + \sigma_{ch,i}^2} \right) , \tag{11}$$

where $\delta_{cj} = 1$ when $c = j$ and $\delta_{cj} = 0$ when $c \neq j$. $\sigma_{ch,i}^2$ is set to zero if code bit i is punctured. Similarly, because $I_{E,V}^{i \to j}$ acts as *a priori* mutual information in the calculation of $I_{E,C}^{j \to i}$, following (8) we have (when $b_{ji} \neq 0$)

$$I_{E,C}^{j \to i} = 1 - J\left(\sqrt{\sum_{v=1}^{N} (b_{jv} - \delta_{ci}) \left(J^{-1}(1 - I_{E,V}^{v \to j}) \right)^2} \right) . \tag{12}$$

The multidimensional EXIT algorithm can now be presented as follows.

1) *Initialization*. Select E_b/N_0. Initialize a vector $\boldsymbol{\sigma}_{ch} = (\sigma_{ch,0}, \ldots, \sigma_{ch,N-1})$ such that
$$\sigma_{ch,i} = 8R \left(\frac{E_b}{N_0} \right)_i$$
where $(E_b/N_0)_i$ equals zero when x_i is punctured and equals the selected E_b/N_0 otherwise.
2) *VN to CN*. For $i = 0, \ldots, N-1$ and $j = 0, \ldots, M-1$, compute (11).
3) *CN to VN*. For $i = 0, \ldots, N-1$ and $j = 0, \ldots, M-1$, compute (12).
4) *Cumulative mutual information*. For $i = 0, \ldots, N-1$, compute
$$I_{CMI}^{i} = J\left(\sqrt{\sum_{c=1}^{M} \left(J^{-1}(I_{E,C}^{c \to i}) \right)^2 + \sigma_{ch,i}^2} \right) .$$
5) If $I_{CMI}^{i} = 1$ (up to desired precision) for all i, then stop; otherwise, go to step 2.

This algorithm converges only when the selected E_b/N_0 is above the threshold. Thus, the threshold is the lowest value of E_b/N_0 for which all I_{CMI}^{i} converge to 1. As shown in [30][31], the thresholds computed by this algorithm are typically within 0.05 dB of those computed by density evolution. Recalling that many classes of multi-edge type (MET) [26] LDPC codes rely on simple protographs, the above algorithm provides an accurate threshold estimation for MET ensembles, with a remarkable reduction in computational complexity relative to the density evolution analysis proposed in [26].

IV. Accumulator-Based Code Designs

A. Repeat-Accumulate Codes

This section provides an overview of the design of LDPC codes that can be considered to be a concatenation of a set

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.