On LDPC Code Ensembles with Generalized Constraints

Yanfang Liu, Pablo M. Olmos, Tobias Koch
Universidad Carlos III de Madrid
& Gregorio Marañón Health Research Institute
Email: {vivian,olmos,koch}@tsc.uc3m.es

Abstract

In this paper, we analyze the tradeoff between coding rate and asymptotic performance of a class of generalized low-density parity-check (GLDPC) codes constructed by including a certain fraction of generalized constraint (GC) nodes in the graph. The rate of the GLDPC ensemble is bounded using classical results on linear block codes, namely Hamming bound and Varshamov bound. We also study the impact of the decoding method used at GC nodes. To incorporate both bounded-distance (BD) and Maximum Likelihood (ML) decoding at GC nodes into our analysis without resorting on multi-edge type of degree distributions (DDs), we propose the probabilistic peeling decoding (P-PD) algorithm, which models the decoding step at every GC node as an instance of a Bernoulli random variable with a successful decoding probability that depends on both the GC block code as well as its decoding algorithm. The P-PD asymptotic performance over the BEC can be efficiently predicted using standard techniques for LDPC codes such as density evolution (DE) or the differential equation method. Furthermore, for a class of GLDPC ensembles, we demonstrate that the simulated P-PD performance accurately predicts the actual performance of the GLPDC code under ML decoding. From the point of view of hardware implementation, the simpler the DDs of GLDPC codes the better. Thus, we illustrate our analysis for GLDPC code ensembles with regular base DDs. In all cases, we show that a large fraction of GC nodes is required to reduce the original gap to capacity. We then consider techniques to further reduce this gap by means of random puncturing, and the inclusion of a certain fraction of generalized variable nodes in the graph.

Index Terms

Generalized low-density parity-check codes, codes on graphs, maximum-likelihood decoding

I. Introduction

Generalized low-density parity-check (GLDPC) block codes were first proposed by Tanner [1]. In contrast to standard LDPC codes, which are represented by bipartite Tanner graphs where variable nodes and single parity-check (SPC) nodes are connected according to a given degree distribution (DD), in GLDPC codes the SPC nodes in the graph are replaced by generalized constraint (GC) nodes [1]. The sub-code associated to each GC node is referred to as the component code. Examples of component codes used in the GLDPC literature are Hamming codes [2], Hadamard codes [3] or expurgated random codes [4], [5]. For more powerful (compared with simple SPC codes) component codes, GLDPC codes have many potential advantages, including improved performance in noisy channels, fast convergence speed and low error floor [4], [6].

Upon selecting a particular class of component codes, the DD of the GLDPC code ensemble can be optimized, and near-capacity iterative decoding thresholds can be achieved [2], [4], [7]. Capacity-achieving GLDPC code ensembles can also be obtained by spatially-coupling GLDPC block codes with regular DDs [8], [9]. Furthermore, the asymptotic exponent of the weight/stopping set spectrum for irregular and spatially-coupled GLDPC ensembles have been derived in [6], [10] respectively. In this regard, it then possible to design asymptotically good GLDPC code ensembles to achieve capacity-approaching iterative decoding thresholds and a minimum distance that grows linearly with the blocklength.

In this paper, we analyze GLDPC code ensembles using a different approach. Instead of selecting a particular class of component codes and optimizing the graph DD, we are interested in analyzing the tradeoff between coding rate and iterative decoding threshold of GLDPC code ensembles with fixed DD, referred to as the base DD, as we increase the fraction ν of GC nodes in the graph. This approach is novel in the literature and we believe it is appealing from a design perspective, since one might be interested in introducing a certain amount of GC nodes in the Tanner graph of a given LDPC code, aiming at reducing the gap to channel capacity at the resulting coding rate, and at the same time improving the minimum distance of the code and thus the error floor.

This work has been funded in part by the Spanish Ministerio de Economía y Competitividad and the Agencia Española de Investigación under Grant TEC2016-78434-C3-3-R (AEI/FEDER, EU) and by the Comunidad de Madrid in Spain under Grant S2103/ICE-2845. T. Koch has further received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 714161), from the 7th European Union Framework Programme under Grant 333680, and from the Spanish Ministerio de Economía y Competitividad under Grants TEC2013-41718-R and RYC-2014-16332. Pablo M. Olmos has further received funding from the Spanish Ministerio de Economía y Competitividad under Grant IJCI-2014-19150.

For the BEC, iterative decoding of graph-based codes, such as LDPC or GLDPC codes, can be performed by means of *peeling decoding* (PD) algorithms [11], [12], [13], which iteratively remove variable nodes whose value is known from the Tanner graph. As a result, the decoding process yields a sequence of graphs whose mean coincides with the asymptotic (in the blocklength) evolution of the ensemble. Furthermore, this evolution can be computed by solving a particular set of differential equations. In the case of GLDPC codes, the derivation of such differential equations requires to specify in advance the DD of the graph, and a description of what kind of erasure patterns are locally decodable at any GC nodes, which depends on both the component codes and the corresponding decoding algorithm. In fact, the resulting decoding threshold of GLDPC codes heavily depends on this latter point [3], [5], [13]. For instance, for a $(2, 7)$-regular base DD in which all check nodes are $(7, 4)$-Hamming GC nodes, the asymptotic threshold over the BEC is $\epsilon^* \approx 0.7025$ if maximum likelihood (ML) decoding is performed at each GC nodes. However, it drops to $\epsilon^* \approx 0.5135$ if suboptimal bounded distance (BD) decoding is used instead of ML. In both cases, the coding rate is exactly the same. While BD-decoded GC nodes only resolve erasure patterns up to degree $\mathtt{d} - 1$, where $\mathtt{d}$ is the minimum distance of the component code, ML-decoded GC nodes can resolve a subset of erasure patterns of degree above $\mathtt{d} - 1$.

While deriving the asymptotic differential equations to analyze PD with BD decoding at GC nodes (BD-PD for short) follows a straightforward extension of the standard PD differential equations for LDPC codes [11]. The GLDPC asymptotic analysis of PD under ML-decoded component codes (ML-PD, for short) requires the use of multi-edge-type DDs [14] to track down all possible decodable erasure patterns at GC nodes [8], [13]. As a consequence, the list of code parameters to jointly optimize becomes cumbersome. In particular, the parameters include the description of the multi-edge DD, the position of GC nodes in the graph, the edge labelling at every GC node used to determine positions in the component block code, and the list of locally ML-decodable erasure patterns. In [5], the authors were able to incorporate ML-decoded GC nodes without resorting on multi-edge type DDs by analyzing the GLDPC average performance using extrinsic information (EXIT) charts when each GC node in the graph is selected at random within the family of block component codes with fixed block length and minimum distance larger than 2. This approach has a design caveat though, as it does not allow the use of a single type of component codes, nor to narrow down the family of component codes by fixing the minimum distance.

In this paper, we propose an analysis methodology that allows to easily incorporate ML-decoded GC nodes with specific properties, such a particular value of the minimum distance $\mathtt{d}$ or how many erasure patterns beyond minimum distance it can decode, into the PD algorithm. We develop a probabilistic description of all components of the GLDPC code, namely the base DD, the presence of GC nodes in the graph, and the decoding method implemented at GC nodes. Regarding the latter aspect, we parameterize

the decoding capabilities of at every blocklength-K component code node by a vector $(p_1, p_2, ..., p_K)$, where $p_w \in [0, 1]$ is the probability that a weight-w erasure pattern chosen at random is decodable, $w \in \{1, \ldots, K\}$. Thus, p_w is the fraction of decodable weight-w erasure patterns. We show how to properly incorporate such a probabilistic description of component codes into the PD algorithm, and denote the resulting algorithm as *probabilistic PD* (P-PD). Due to its probabilistic nature, the asymptotic analysis of P-PD does not require the use of multi-edge type DDs. By computer simulations as we ensure the edge labelling at the GC nodes is chosen uniformly at random, we show that the P-PD performance accurately predicts the actual GLDPC performance when ML decoding is performed at GC nodes. Note that if we take $p_w = 1$ for $w \leq \mathtt{d} - 1$ and $p_w = 0$ for $w = \{\mathtt{d}, \ldots, K\}$, we recover BD-PD.

Note that the performance predicted using P-PD is valid for the family of linear blocklength-K block component codes with the same decoding profile $(p_1, p_2, ..., p_K)$. In this regard, we propose two bounds on the GLDPC code rate for this family of linear block component codes. The Hamming or sphere-packing bound [15] is used to determine a converse bound on the rate of the GLDPC code ensemble as a function of a triplet of $(\nu, \mathtt{d}, K)$. The Varshamov bound is considered to determine an achievable rate of the GLDPC code ensemble [16]. In many scenarios of interest, we show that these bounds are tight, and thus relevant for the code designer.

By exploiting the probabilistic description of the decoding capabilities at GC nodes, we are able to analyze a large class of GLDPC code ensembles and beyond-BD decoding methods with a fairly small set of parameters. We illustrate our analysis for GLDPC code ensembles using $(2, 6)$, $(2, 7)$, $(2, 8)$ and $(2, 15)$ base DDs. We have performed an exhaustive search of linear block codes of lengths $r \in [6, 7, 8, 15]$, including Hamming codes, Cyclic codes, Quasi Cyclic codes and Cordaro-Wagner Codes, and tabulated their corresponding description in terms of minimum distance $\mathtt{d}$ and $(p_1, p_2, ..., p_K)$ parameters. In all cases, we show that a large fraction of GC nodes is required in the GLDPC graph to approximately maintain or reduce the original gap to capacity. However, the closest gap to capacity is not achieved at $\nu = 1$, but a smaller value must be used. Namely, there exists a critical ν^* value for which the gap to capacity is minimum. Furthermore, the best results are obtained for high-rate component codes, suggesting that the use of very powerful component codes does not pay off. The gain in threshold does not compensate for the severe decrease of the GLDPC code rate. We then combine the asymptotic threshold results with the weight spectral analysis of GLDPC ensemble in [10] and explore the range of ν values for which the GLDPC ensembles reduce the original gap to capacity and at the same time maintain a linear growth of the minimum distance with the block length. Finally, we illustrate how to incorporate further design techniques that can help to reduce the gap to capacity of the code ensembles. We consider both random puncturing [17] and a simple class doubly generalized LDPC (DG-LDPC) codes [18], [19].

The methodology presented is flexible and decouples the problems of bounding the GLDPC coding rate and the asymptotic analysis of the ensemble. In this regard, broader classes of component codes at variable nodes and GC nodes can be incorporated in a systematic way. The paper is organized as follows. In Section II, we introduce GLDPC code ensembles and the notation used to characterize the DDs. Section III presents the decoding algorithm and its asymptotic analysis. In Section V we bound the GLDPC code rate and analyze the rate-threshold tradeoff as a function of the fraction ν of GC nodes in the graph. The behavior of the GLDPC code ensembles with specific component codes is analyzed in VI. Finally, Sections VII and VIII consider further techniques to improve the asymptotic behavior of the code ensemble, by means of random puncturing and generalized variable nodes. We finish the paper in Section IX with a summary of our conclusions and some insight into future lines of research.

II. GLDPC ENSEMBLES

In this section, we introduce the GLDPC code ensembles that will be analyzed in the rest of the paper and the notation used to define their DD.

A. Degree distribution

As illustrated in Fig. 1, the Tanner graph of every member in the ensemble contains $\mathtt{n}$ variable nodes (coded bits) and $\mathtt{c}$ parity-check nodes, among which a fraction ν corresponds to GC nodes while the rest corresponds to SPC nodes. We denote by $\mathtt{E}$ the number of edges in the Tanner graph and we define the degree of a node as the number of edges connected to it.

The DD of the ensemble is characterized as follows. The vector $\overline{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_J)$ is the *left* DD, where λ_i represents the fraction of edges (w.r.t. $\mathtt{E}$) connected to a variable of degree i. Given $\overline{\lambda}$, $\mathtt{n}$ and $\mathtt{E}$ are related by [14]

$$\mathtt{n} = \mathtt{E} \sum_{i=1}^{J} \lambda_i / i. \tag{1}$$

The *right* DD is defined by two vectors $\overline{\rho}_p = (\rho_{p1}, \rho_{p2}, ..., \rho_{pK})$ and $\overline{\rho}_c = (\rho_{c1}, \rho_{c2}, ..., \rho_{cK})$, where ρ_{pj} denotes the fraction of edges (w.r.t. $\mathtt{E}$) connected to a SPC node that has degree j and ρ_{cj} denotes the fraction of edges (w.r.t. $\mathtt{E}$) connected to a GC node that has degree j. Since the fraction of GC nodes in the graph is ν, the following must hold:

$$\nu = \frac{\sum_{j=1}^{K} \rho_{cj}/j}{\sum_{u=1}^{K} (\rho_{cu} + \rho_{pu})/u}. \tag{2}$$

For simplicity, we restrict our analysis to the class of GLDPC ensembles characterized by SPC and GC nodes with constant degree J and K, respectively. The Tanner graph of any code in this ensemble

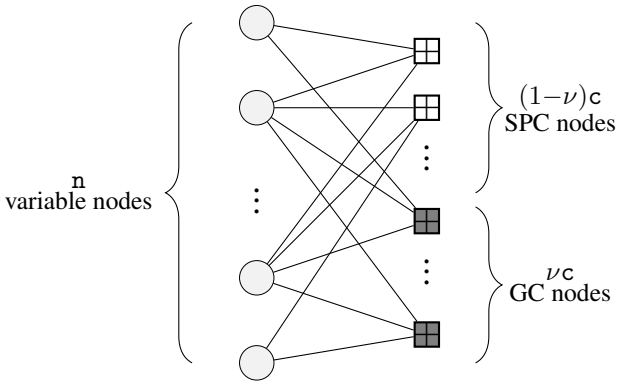


Fig. 1. Tanner graph of a GLDPC code.

contains $\mathtt{n}$ variable nodes, $\mathtt{E} = J\mathtt{n}$ edges, $\nu\frac{J}{K}\mathtt{n}$ GC nodes, and $(1-\nu)\frac{J}{K}\mathtt{n}$ SPC nodes. The DD of the GLDPC codes is characterized by the triple (J, K, ν), and the ensemble of codes generated by this DD is denoted by $\mathcal{C}_{J,K,\nu}$. The DD of the LDPC ensemble obtained by taking $\nu = 0$ is defined as the *base DD*, and the corresponding LDPC code ensemble is referred to as the *base ensemble*. The coding rate of the base ensemble is denoted by $\mathtt{R}_0$ and can be computed as:

$$\mathtt{R}_0 = 1 - \frac{J}{K}. \tag{3}$$

Finally, we assume that the incoming edges to every degree-K GC nodes are assigned uniformly at random to each position of the component code.

B. The coding rate of the $\mathcal{C}_{J,K,\nu}$ ensemble

As discussed in the introduction of the paper, we propose tools to analyze the decoding performance of GLDPC under ML-decoded GC nodes that do not require to set in advance a specific component code to be used as the GC nodes. In this regard, we consider the family of linear block codes with blocklength K and minimum distance $\mathtt{d}$ and we use the classical results on linear block codes to find bounds on the coding rate of the GLDPC code ensembles.

Let $\mathtt{k}^{(\ell)} \in \mathbb{N}^+$, $\ell = 1, \ldots, \nu\mathtt{E}/K$, be the number of rows in the parity-check matrix associated with the component code of the ℓ-th GC node.

Lemma 2.1: The design rate $\mathtt{R}(\nu)$ of the $\mathcal{C}_{J,K,\nu}$ ensemble is

$$\mathtt{R}(\nu) = \mathtt{R}_0 - \nu(1-\mathtt{R}_0)(\mathtt{k}_{\text{avg}} - 1), \tag{4}$$

where $\mathtt{k}_{\text{avg}} = (\nu\frac{\mathtt{E}}{K})^{-1}\sum_{\ell=1}^{\nu\frac{\mathtt{E}}{K}}\mathtt{k}^{(\ell)}$ is the average number of rows in the parity-check matrix of the component codes.

Proof: Any SPC node in the Tanner graph accounts for a single row in the parity-check matrix of the GLDPC code, and any GC node accounts for $\mathtt{k}^{(\ell)}$ rows. Thus, the design rate $\mathtt{R}(\nu)$ is given by

$$\mathtt{R}(\nu) = 1 - \frac{(1-\nu)\frac{\mathtt{E}}{K} + \sum_{\ell=1}^{\nu\frac{\mathtt{E}}{K}} \mathtt{k}^{(\ell)}}{\mathtt{n}} = 1 - \frac{(1-\nu)\frac{\mathtt{E}}{K} + \nu\frac{\mathtt{E}}{K}\mathtt{k}_{\text{avg}}}{\mathtt{E}/J} = \mathtt{R}_0 - \nu(1-\mathtt{R}_0)(\mathtt{k}_{\text{avg}} - 1). \quad (5)$$

■

Note that the second term in (4) accounts for the rate loss at GC nodes. The fact that component codes are linear block codes with minimum distance $\mathtt{d}$ can be used to derive the following bounds on the $\mathtt{R}(\nu)$:

Lemma 2.2: If all component codes in the $\mathcal{C}_{J,K,\nu}$ ensemble are linear block codes with minimum distance $\mathtt{d} > 2$, then

$$\mathtt{R}(\nu) \leq \mathtt{R}_0 - \nu(1-\mathtt{R}_0)\log_2\left(\frac{1}{2}\sum_{q=0}^{\lfloor\frac{\mathtt{d}-1}{2}\rfloor}\binom{K}{q}\right). \quad (6)$$

Furthermore, there exists a set of linear block codes to be used as component codes such that

$$\mathtt{R}(\nu) \geq \mathtt{R}_0 - \nu(1-\mathtt{R}_0)\left\lceil\log_2\left(\frac{1}{2} + \frac{1}{2}\sum_{q=0}^{\mathtt{d}-2}\binom{K-1}{q}\right)\right\rceil. \quad (7)$$

Here, we use $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ to denote the ceiling and floor functions, respectively. The two bounds coincide, for example, when $\mathtt{d} = 3$ and $r = 2^z - 1$, where $z \in \mathbb{Z}_+$. Thus, in this case the bound in (7) is achieved with equality.

Proof: First, the condition $\mathtt{d} > 2$ is required to differentiate between the rate loss at SPC nodes, which are block codes with minimum distance 2, and at GC nodes. We start by proving the converse bound in (6). By the sphere-packing bound in [15] [Theorem 12, page 531], any component code with blocklength K and minimum distance $\mathtt{d}$ must satisfy

$$2^{K-\mathtt{k}} \leq \frac{2^K}{\sum_{q=0}^{\lfloor\frac{\mathtt{d}-1}{2}\rfloor}\binom{K}{q}}, \quad (8)$$

where $\mathtt{k}$ is the number of rows in the parity-check matrix. This implies that the term $(\mathtt{k}_{\text{avg}} - 1)$ in (4) is bounded by

$$\mathtt{k}_{\text{avg}} \geq \log_2\left(\sum_{q=0}^{\lfloor\frac{\mathtt{d}-1}{2}\rfloor}\binom{K}{q}\right), \quad (9)$$

which proves (6). Regarding the achievable bound in (7), the Varshamov Bound [16, Theorem 2.9.3] guarantees the existence of a linear component code with blocklength K and minimum distance at least $\mathtt{d}$ if

$$2^{K-\mathtt{k}} \geq 2^{\mathtt{n}-\left\lceil\log_2\left(1+\sum_{q=0}^{\mathtt{d}-2}\binom{K-1}{q}\right)\right\rceil}. \quad (10)$$

If the above condition is satisfied, there exists a set of linear block codes to be used as component codes with blocklength K and minimum distance at least $\mathtt{d}$ such that

$$\mathtt{k}_{\text{avg}} - 1 \leq \left\lceil \log_2 \left(\frac{1}{2} + \frac{1}{2} \sum_{q=0}^{\mathtt{d}-2} \binom{K-1}{q} \right) \right\rceil, \tag{11}$$

which proves (7).

Finally, if we substitute $\mathtt{d} = 3$ and $K = achievable 2^z - 1$ for some $z \in \mathbb{Z}_+$ into (6) and (7), a straightforward computation shows that the converse bound in (6) can be simplified to

$$\mathtt{R}(\nu) \leq \mathtt{R}_0 - \nu(1 - \mathtt{R}_0)(z-1), \tag{12}$$

and, similarly, the achievable bound in (7) simplifies to

$$\mathtt{R}(\nu) \geq \mathtt{R}_0 - \nu(1 - \mathtt{R}_0)(z-1). \tag{13}$$

■

C. Growth Rate of the Weight Distribution for the $\mathcal{C}_{J,K,\nu}$ ensemble

A useful tool for analysis and design of LDPC codes and their generalizations is represented by the asymptotic exponent of the weight distribution. The growth rate of the weight distribution was introduced in [20] to show that the minimum distance of a randomly generated regular LDPC code with a VN degree of at least three is a linear function of the codeword length with high probability. The growth rate of the weight distribution for a class of doubly generalized LDPC (D-GLDPC) codes was introduced in [10]. The $\mathcal{C}_{J,K,\nu}$ GLDPC code ensemble can be seen as a particular instance of the codes analyzed in that work. The weight spectral shape of the $\mathcal{C}_{J,K,\nu}$ ensemble captures the behavior of codewords whose weight is linear in the block length $\mathtt{n}$ and is defined by

$$G(\alpha) \triangleq \lim_{\mathtt{n}\to\infty} \frac{1}{\mathtt{n}} \log \mathbb{E}_{\mathcal{C}_{J,K,\nu}}[A_{\alpha\mathtt{n}}] \tag{14}$$

for $\alpha > 0$. This limit assumes the inclusion of only those positive integers for which $\alpha\mathtt{n} \in \mathbb{Z}$. We define the *critical exponent codeword weight ratio* as $\hat{\alpha} \triangleq \inf\{\alpha \geq 0 | G(\alpha) \geq 0\}$. If $\hat{\alpha} > 0$, then the code's minimum distance asympotitically grows as $\mathcal{O}(\hat{\alpha}\mathtt{n})$ and the ensemble is said to have good growth rate behavior. Note that, even though $\hat{\alpha} = 0$, the minimum distance of the code may still grow with the block length $\mathtt{n}$ but at a slower rate, e.g., as the $\log(\mathtt{n})$.

Lemma 2.3: If all component codes in the $\mathcal{C}_{J,K,\nu}$ ensemble are linear block codes with minimum distance $\mathtt{d} > 2$, then $\hat{\alpha} > 0$ for $J > 2$. For $J = 2$, $\hat{\alpha} > 0$ if and only if

$$\nu > \frac{K-2}{K-1} \triangleq \hat{\nu}. \tag{15}$$

Otherwise, $\hat{\alpha} = 0$.

Proof: The proof is straightforward by particularizing the results in [10] [Section II] to the $\mathcal{C}_{J,K,\nu}$ ensemble. ■

III. Probabilistic Peeling Decoding over the BEC

Suppose we use a random sample of the $\mathcal{C}_{J,K,\nu}$ ensemble to transmit over a BEC(ϵ). For this channel, each of the $\mathtt{n}$ coded bits is erased with probability ϵ. Without loss of generality, we assume that the all-zero codeword is transmitted, hence the received vector $\boldsymbol{y}$ belongs to the set $\{0, ?\}^{\mathtt{n}}$, where $?$ denotes an erasure. Let $\Gamma_{\boldsymbol{y}} \subseteq \{1, \ldots, \mathtt{n}\}$ be the index set of the bits correctly received, namely $y_i = 0$ for all $i \in \Gamma_{\boldsymbol{y}}$. Decoding will be performed using a generalization of the PD algorithm [11] similar to that proposed for GLDPC codes in [13]. The final formulation of the decoding algorithm depends on the decoding capabilities we assume at GC nodes. For instance, if we assume BD decoding at component codes, then the generalized PD algorithm, denoted as BD-PD, proceeds as described in Algorithm 1.

Algorithm 1 BD-PD

Remove from the Tanner graph of the GLDPC code all variable nodes with indexes in $\Gamma_{\boldsymbol{y}}$.

Construct Ψ, the index set of check nodes that correspond to either degree-one SPC nodes or GC nodes of degree less or equal to $\mathtt{d} - 1$.

repeat

1) Select at random a member of Ψ.

2) Remove from the Tanner graph the check node with the index drawn in Step 1). Further, remove all connected variable nodes, and all attached edges.

3) Update Ψ.

until All variable nodes have been removed (successful decoding) or $\Psi = \emptyset$ (decoding failure).

BD-PD is a suboptimal decoding method that considers decodable all GC nodes up to degree $\mathtt{d} - 1$ [9], [21], thus it ignores the fact that any component code will be able to decode a certain fraction of erasure patterns of weight equal to or greater than $\mathtt{d}$. As already reported in many works, e.g., [8], [13], the GLPDC code performance dramatically improves if we consider ML decoding at GC nodes. In principle, to consider ML decoding at GC nodes, we have to specify a full list of decodable erasure patterns and, consequently, label each of the incoming edges at every GC node to differentiate between decodable and non-decodable GC nodes. As shown in [13], incorporating this labelling into the asymptotic analysis requires the use of multi-edge type DDs.

Algorithm 2 P-PD

Remove from the Tanner graph of the GLDPC code all variable nodes with indexes in $\Gamma_{\boldsymbol{y}}$.

for all GC nodes **do**

If the GC has degree w, tag the check node as *decodable* with probability p_w.

end for

Construct Ψ, the index set of check nodes corresponding to either degree-one SPC nodes or GC nodes tagged as decodable.

repeat

1) Select at random a member of Ψ.

2) Remove from the Tanner graph the check node with the index drawn in Step 1). Further, remove all connected variable nodes, and all attached edges.

3)

for every non-decodable GC node that has lost one or more edges **do**

If the GC has degree w, tag the check node as *decodable* with probability p_w.

end for

4) Update Ψ.

until All variable nodes have been removed (sucessful decoding) or $\Psi = \emptyset$ (decoding failure).

In order to incorporate beyond-BD decoding at GC nodes into our analysis, and at the same time maintain a formulation compatible with the random definition of the $\mathcal{C}_{J,K,\nu}$ ensemble, we will further constrain the family of component codes to be used at degree-K GC nodes. More specifically, we assume that the fraction of ML-decodable weight-w erasure patterns at every GC node is given by some $p_w \in [0,1], w = 1, \ldots, K$. Thus, the family of component codes under analysis is the family of blocklength-K linear block codes with minimum distance $\mathtt{d}$ and with decoding profile described by the vector $\mathbf{p} = (p_1, \ldots, p_K)$. Note that if minimum distance of the component code is $\mathtt{d}$, then $p_w = 1$ for $w \leq \mathtt{d} - 1$.

By exploiting the fact that incoming edges at every GC node are uniformly at random assigned to each position of the component code, we can incorporate ML-decoded GC nodes into PD as shown in Algorithm 2, denoted as probabilistic PD (P-PD). Observe that the key P-PD feature is tagging GC check nodes as decodable with probabilities given by $\mathbf{p}$ only when they lose one or more edges, which may happen either at an initialization or after a connected variable is removed. If only a decodable check node is removed per iteration, only a few GC nodes are can change its state (from non-decodable to

decodable) after every P-PD iteration. Thus, at every iteration, P-PD emulates the ML decoding operation of a degree-w GC node by drawing the decoding capability according to a Bernoulli distribution with parameter p_w, $w \in \{1, \ldots, K\}$. Note that P-PD must be regarded as a procedure that allows for simpler analysis rather than a practical decoding algorithm. Further, note that we recover the bounded distance PD (BD-PD) algorithm from P-PD if we set $p_w = 0$ for $w \geq \mathtt{d}$ and $p_w = 1$ otherwise.

Finally, note that the bounds on $\mathtt{R}(\nu)$, predicted in Lemma 2.3, could in principle be refined according to $\mathbf{p}$. While this is an interesting open question, we will later show that the bounds are tight in certain scenarios and there is little room for refinement.

A. Comparing the P-PD and ML-PD performance by Monte Carlo simulation

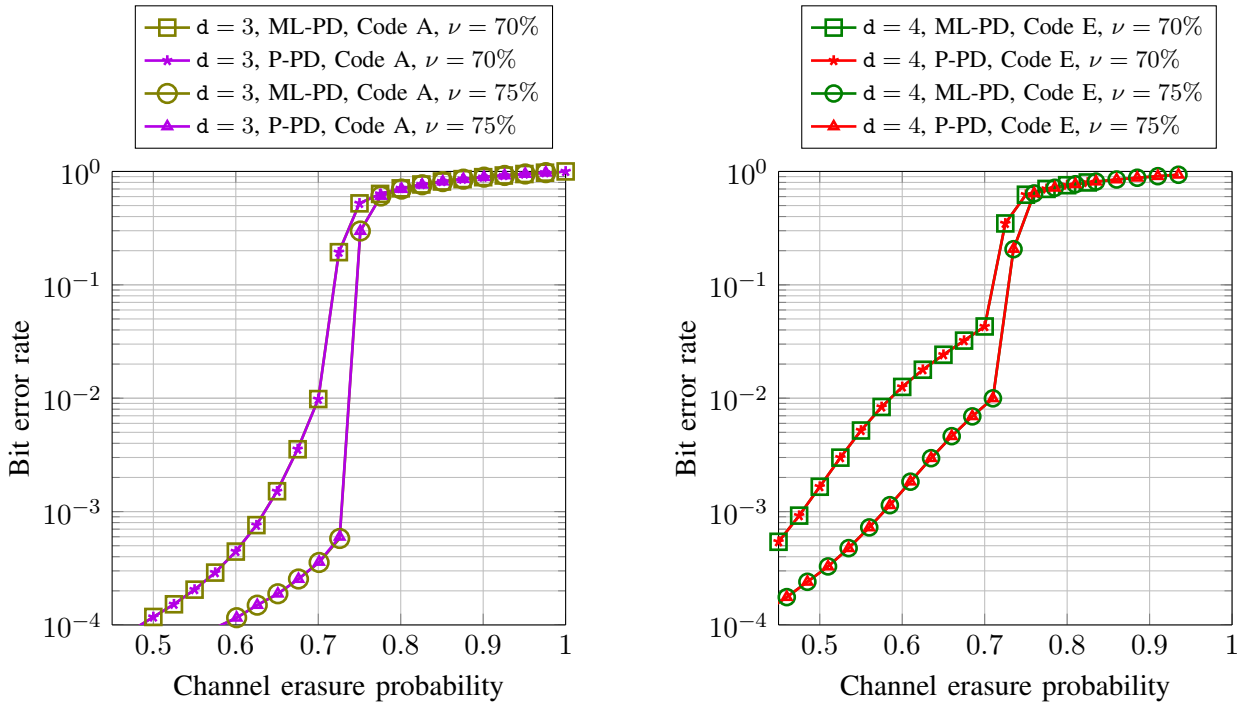
If we select a specific block component code, we can compare the simulation performance of the $\mathcal{C}_{J,K,\nu}$ ensemble based on corresponding parameters under P-PD with that of the practical GLDPC codes with GC nodes that are decoded via ML, using the actual parity-check matrix of the component codes. We refer to this latter case as ML-PD. More precisely, for a given finite blocklength $\mathtt{n}$, fixed $\nu \in [0, 1]$ and base DD, we generate a member of the $\mathcal{C}_{J,K,\nu}$ ensemble as follows

1) Draw at random a member of the $\mathcal{C}_{J,K,\nu}$ ensemble of GLDPC codes, which generates a Tanner graph with $\mathtt{n}$ nodes, $\nu\mathtt{E}/K$ GC nodes and $(1-\nu)\mathtt{E}/K$ SPC nodes.
2) For each of the $\nu\mathtt{E}/K$ GC nodes, we generate uniformly at random a permutation of the set $\{1, 2, \ldots, K\}$, which is used to associate each of the incomming edges to the GC node to a position in the block component code.

Then, we estimate by Monte Carlo simulation the bit error rate (BER) over the BEC achieved by both P-PD, which does not use the labelling of edges per GC node but rather follows Algorithm 2, and ML-PD, which uses the labelling of edges per GC node to create a look-up table of decodable erasure patterns. We show results in Fig. 2 (a) for a $(2, 6)$-regular base DD with a rate-$1/2$ Hamming $(6, 3)$ linear block code as component code. In Fig. 2 (b) we consider for a $(2, 8)$-regular base DD using a rate-$1/2$ $(8, 4)$ Hamming component code. Results have been averaged with 10 generated samples from the $\mathcal{C}_{J,K,\nu}$ ensemble. Observe the perfect match between both decoders in all cases. This shows we are not sacrificing accuracy with the probabilistic description of the decoder, as long as GLDPC codes are generated as described above.

IV. Asymptotic analysis

The P-PD decoder yields a sequence of residual graphs by sequentially removing degree-one SPC nodes and decodable GC nodes from the GLDPC Tanner graph. Our next goal is to predict the asymptotic



(a) (b)

Fig. 2. In (a), we plot simulation results using a $(2, 6)$-regular base DD with a rate-$1/2$ Hamming $(6, 3)$ linear block code as component code. In (b), we consider a $(2, 8)$-regular base DD using a rate-$1/2$ $(8, 4)$ Hamming component code. Results have been averaged with 10 generated samples from the $\mathcal{C}_{J,K,\nu}$ ensemble with a blocklength of 10000 bits.

behaviour of the $\mathcal{C}_{J,K,\nu}$ ensemble under P-PD by extending the methodology proposed in [11] to analyze the asymptotic behavior of LDPC ensembles under PD. In [11], it is shown that if we apply the PD to elements of an LDPC ensemble, then the expected DD of the sequence of residual graphs can be described as a solution of a set of differential equations. Furthermore, the deviation of the process w.r.t. the expected evolution decreases exponentially fast with the LDPC blocklength. This analysis is based on a result on the evolution of Markov processes due to Wormald [22]. We first review this result, and then prove that the GLDPC graph evolution under P-PD satisfies its conditions.

A. Wormald's theorem [22]

Let $\{Z^{(\ell)}(a), \ell \in \mathbb{N}_+\}$ be a d-dimensional discrete-time Markov random process with state space $\{0, 1, \ldots, \lfloor a\alpha \rfloor\}^d$ for some parameters $\alpha > 0$ and $a > 1$, where ℓ denote the time index. Further let $Z_i^{(\ell)}(a)$ denote the i-th component of $Z^{(\ell)}(a)$, $i = 1, \ldots, d$. Let D be a subset of $\mathbb{R}^{d+1}$ containing those

vectors $[0, z_1, ..., z_d]$ such that:

$$\mathbb{P}\left(\frac{Z_i^{(0)}(a)}{a} = z_i, 1 \leq i \leq d\right) > 0. \tag{16}$$

We define the stopping time ℓ_D to be the smallest time index ℓ such that

$$(\ell_D/a, Z_1^{(\ell)}(a)/a, ..., Z_d^{(\ell)}(a)/a) \notin D \tag{17}$$

Furthermore, let $f_i(\cdot)$, $i = 1, \ldots, d$, be functions from $\mathbb{R}^{d+1}$ to $\mathbb{R}$ such that the following conditions are satisfied:

1) (Boundedness) There exists a constant Ω such that for all $i = 1, \ldots d$, $\ell = 0, \ldots, \ell_D - 1$ and $a \geq 1$,

$$\left|Z_i^{(\ell+1)}(a) - Z_i^{(\ell)}(a)\right| \leq \Omega,$$

2) (Trend functions) For all $i = 1, \ldots, d$, $\ell = 0, \ldots, \ell_D - 1$ and $a \geq 1$,

$$\mathbb{E}\left[Z_i^{(\ell+1)}(a)/a - Z_i^{(\ell)}(a)/a \middle| Z^{(\ell)}(a)/a\right] = f_i\left(\ell/a, Z_1^{(\ell)}(a)/a, ..., Z_d^{(\ell)}(a)/a\right) + \mathcal{O}(1/a),$$

3) (Lipschitz continuity) Each function $f_i(\cdot)$, $i = 1, \ldots, d$, is Lipschitz continuous on the intersection between D and the half space $[\tau, z_1, \ldots, z_d]$ with $\tau > 0$. Namely, for any pair $b, c \in \mathbb{R}^{d+1}$ that belongs to such intersection, there exists a constant κ such that

$$|f_i(b) - f_i(c)| \leq \kappa \sum_{j=1}^{d+1} |b_j - c_j|.$$

Under these conditions, the following holds:

- The system of differential equations

$$\frac{\partial z_i}{\partial \tau} = f_i(\tau, z_1, ..., z_d), \quad i = 1, ..., d, \tag{18}$$

  has a unique solution for any initial condition $[0, b_1, ..., b_d] \in D$.

- There exists a strictly positive constant ζ such that

$$\mathbb{P}\left(\left|Z_i^{(\ell)}(a)/a - z_i(\ell/a)\right| > \zeta a^{-\frac{1}{6}}\right) = \mathcal{O}\left(\mathrm{e}^{-\sqrt{a}}\right) \tag{19}$$

  for $i = 1, ..., d$ and $0 \leq t \leq t_D$, where $z_i(\ell/a)$ is the solution given by equation (18) for

$$b_i = \mathbb{E}[Z_i^{(0)}(a)]/a. \tag{20}$$

The result in (19) states that any realization of the process $Z_i^{(t)}(a)$ concentrates around the solution predicted by (18) in the limit $a \to \infty$. In the next subsecion we show that this theorem is suitable to describe the expected GLDPC graph evolution during P-PD decoding process.

B. Expected graph evolution under P-PD

We first introduce the notation used to characterize the DD of the ensemble of residual Tanner graphs. The key aspect is to augment the DD notation introduced in Section II to differenciate between GC nodes that have been tagged as decodable and those tagged as non-decodable. In order to simplify the formulation, we restrict ourselves to the case $p_w = 0$ for $w \geq \mathtt{d}+2$, i.e., we consider component codes can only decode a certain fraction of erasure patterns of degrees $\mathtt{d}$ and $\mathtt{d}+1$ and all erasure patterns of degree below $\mathtt{d}$. This may not be an strong assumption. After exhaustive search of short linear block component codes (blocklengths up to 15 bits), we have not found any component code with $p_w > 0$ for $w \geq \mathtt{d}+2$. In any case, the extension of the analysis provided here to the general case is straightforward.

Recall that, as introduced in Section II, any edge adjacent to a degree i variable node is said to have left degree i. Similarly, any edge adjacent to a degree j SPC (GC) node is said to have right SPC (GC) degree j. Given the residual graph at the ℓ-th iteration of the P-PD algorithm, let $L_i^{(\ell)}$ denote the number of edges with left degree i, $i = 1, ..., J$. Similarly, let $R_{pj}^{(\ell)}$ denote the number of edges with right SPC degree j and $R_{cj}^{(\ell)}$ denote the number of edges with right GC degree j at iteration ℓ, $j = 1, ..., K$. We split $R_{cj}^{(\ell)}$ for $j \in \{\mathtt{d}, \mathtt{d}+1\}$ into two terms. Specifically, for $j \in \{\mathtt{d}, \mathtt{d}+1\}$, let $\hat{R}_{cj}^{(\ell)}$ denote the number of edges with right GC degree j connected to GC nodes tagged as decodable, and let $\bar{R}_{cj}^{(\ell)}$ denote the number of edges with right GC degree j connected to GC nodes tagged as not-decodable, we have $R_{cj}^{(\ell)} = \hat{R}_{cj}^{(\ell)} + \bar{R}_{cj}^{(\ell)}, j = \mathtt{d}, \mathtt{d}+1$.

Thus, the Markov random process $\{Z^{(\ell)}(a), \ell \in \mathbb{N}_+\}$ in the previous section is given by the random process $\{\mathcal{G}^{(\ell)}, \ell \in \mathbb{N}_+\}$, where

$$\{\mathcal{G}^{(\ell)}, \ell \in \mathbb{N}_+\} = \{L_i^{(\ell)}, R_{pj}^{(\ell)}, R_{cj}^{(\ell)}, \hat{R}_{c\mathtt{d}}^{(\ell)}, \bar{R}_{c\mathtt{d}}^{(\ell)}, \hat{R}_{c(\mathtt{d}+1)}^{(\ell)}, \bar{R}_{c(\mathtt{d}+1)}^{(\ell)}\} \tag{21}$$

where $i = 1, \ldots, J, j = 1, \ldots, \mathtt{d}-1, \mathtt{d}+2, \ldots, K$. Any component in $\mathcal{G}^{(\ell)}$ belongs to the set $\{0, 1, \ldots, \mathtt{E}\}$. Recall that $\mathtt{E}$ denotes the number of edges in the original GLPDC graph, and thus $\mathtt{E}$ will play the role of parameter a. We define the following normalized quantities:

$$\tau \triangleq \frac{\ell}{\mathtt{E}}, \quad l_i^{(\ell)} \triangleq \frac{L_i^{(\ell)}}{\mathtt{E}}, \quad r_{pj}^{(\ell)} \triangleq \frac{R_{pj}^{(\ell)}}{\mathtt{E}}, \quad r_{cj}^{(\ell)} \triangleq \frac{R_{cj}^{(\ell)}}{\mathtt{E}}, \quad \hat{r}_{c\omega}^{(\ell)} \triangleq \frac{\hat{R}_{c\omega}^{(\ell)}}{\mathtt{E}}, \quad \bar{r}_{c\omega}^{(\ell)} \triangleq \frac{\bar{R}_{c\omega}^{(\ell)}}{\mathtt{E}},$$

for $i \in \{1, \ldots, J\}, j \in \{1, \ldots, K\}$ and $w \in \{\mathtt{d}, \mathtt{d}+1\}$. We have that

$$r_{cw}^{(\ell)} = \hat{r}_{c\omega}^{(\ell)} + \bar{r}_{c\omega}^{(\ell)}, \quad w = \mathtt{d}, \mathtt{d}+1, \tag{22}$$

$$e(\ell) \triangleq \sum_{i=1}^{J} l_i^{(\ell)} = \sum_{j=1}^{K} [r_{pj}^{(\ell)} + r_{cj}^{(\ell)}], \tag{23}$$

and $e(\ell)$ is the fraction of edges remaining in the residual graph at time ℓ. The P-PD process starts at $\ell = 0$, after BEC transmission and initialization. The following relation holds between the quantities defined above at $\ell = 0$ and the $\mathcal{C}_{J,K,\nu}$ DD described in Section II:

$$\mathbb{E}[l_i^{(0)}] = \epsilon\lambda_i, \tag{24}$$

$$\mathbb{E}[r_{pj}^{(0)}] = \sum_{\alpha\geq j} \rho_{p\alpha} \binom{\alpha-1}{j-1} \epsilon^j (1-\epsilon)^{\alpha-j}, \tag{25}$$

$$\mathbb{E}[r_{cj}^{(0)}] = \sum_{\alpha\geq j} \rho_{c\alpha} \binom{\alpha-1}{j-1} \epsilon^j (1-\epsilon)^{\alpha-j}, \tag{26}$$

for $i = 1, \dots J$ and $j = 1, \dots, K$, where the expectation is computed w.r.t. the $\mathcal{C}_{J,K,\nu}$ ensemble and the channel output. Upon initialization, every degree-$\mathtt{d}$ GC node is tagged as decodable with probability $p_{\mathtt{d}}$. Similarly with degree-$\mathtt{d}+1$ GC nodes. We thus have the following initial conditions

$$\begin{aligned} &\mathbb{E}[\hat{r}_{c\omega}^{(0)}] = p_\omega \mathbb{E}[r_{c\omega}^{(0)}], \\ &\mathbb{E}[\bar{r}_{c\omega}^{(0)}] = (1-p_\omega)\mathbb{E}[r_{c\omega}^{(0)}], \quad \omega = \mathtt{d}, \mathtt{d}+1. \end{aligned} \tag{27}$$

Note that all (24)-(27) correspond to the initial conditions in (20). To analyze the behaviour of the P-PD algorithm, assume we observe $\mathcal{G}^{(\ell)}$. To derive the conditional expectations in condition 2) of Wormald's Theorem, the so-called trend functions, we have to average among every possible scenario that we can observe after the P-PD iteration. According to step 1) in Algorithm 2, we chose at random a decodable check node. Let P_{p1} be the probability of selecting a degree-one SPC node, and let P_{cj} denote the probability of selecting a degree-j GC node, $j = 1, \dots, \mathtt{d}+1$. By a simple counting argument, if the check node is selected uniformly at random then

$$P_{p1} = \frac{r_{p1}^{(\ell)}}{r_{\text{sum}}}, \tag{28}$$

$$P_{cj} = \frac{r_{cj}^{(\ell)}/j}{r_{\text{sum}}} \quad j < \mathtt{d}, \tag{29}$$

$$P_{cj} = \frac{\hat{r}_{cj}^{(\ell)}/j}{r_{\text{sum}}} \quad j \in \{\mathtt{d}, \mathtt{d}+1\}. \tag{30}$$

Where $r_{\text{sum}} = r_{p1}^{(\ell)} + \sum_{w=1}^{\mathtt{d}-1} r_{cw}^{(\ell)}/w + \hat{r}_{c\mathtt{d}}^{(\ell)}/\mathtt{d} + \hat{r}_{c(\mathtt{d}+1)}^{(\ell)}/(\mathtt{d}+1)$ is the normalized sum of decodable check nodes at the ℓ-th iteration. Our goal is to derive the evolution of the expected values of $\{L_i^{(\ell)}, R_{pj}^{(\ell)}, R_{cj}^{(\ell)}, \hat{R}_{c\omega}^{(\ell)}, \bar{R}_{\omega}^{(\ell)}\}$. A detailed explanation is given in Appendix A. The conditional expected change of the edges with left degree i, conditional on the graph can be written as

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = -\frac{i l_i^{(\ell)}}{e(\ell)} \left(P_{p1} + \sum_{w=1}^{\mathtt{d}+1} w P_{cw}\right) + \mathcal{O}(1/\mathtt{E}) \tag{31}$$

$$\triangleq f_i(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E}). \tag{32}$$

where note that the function $f_i(\mathcal{G}^{(\ell)}/\mathtt{E})$ depends on all terms in the process $\mathcal{G}^{(\ell)}$, normalized by $\mathtt{E}$. Conditioned on $\mathcal{G}^{(\ell)}$, The expected change in the number of edges with right SPC degree j, is given by

$$\begin{aligned}
&\mathbb{E}\left[R_{pj}^{(\ell+1)} - R_{pj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}\right] \\
&= P_{p1}\left((r_{p(j+1)}^{(\ell)} - r_{pj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} - \mathbb{I}[j=1]\right) \\
&+ \sum_{w=1}^{\mathtt{d}+1} P_{cw}(r_{p(j+1)}^{(\ell)} - r_{pj}^{(\ell)})\frac{jw(a(\ell)-1)}{e(\ell)} + \mathcal{O}(1/\mathtt{E}) \\
&\triangleq g_{pj}(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E}),
\end{aligned} \tag{33}$$

where $a(\ell) = \sum_i i l_i^{(\ell)}/e(\ell)$ is the average left degree. Similarly, the conditional expected change in the number of edges with right GC node degree $j \notin \{\mathtt{d}, \mathtt{d}+1\}$ is

$$\begin{aligned}
&\mathbb{E}\left[R_{cj}^{(\ell+1)} - R_{cj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}\right] \\
&= P_{p1}\left((r_{c(j+1)}^{(\ell)} - r_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)}\right) \\
&+ \sum_{w=1}^{\mathtt{d}+1} P_{cw}\left((r_{c(j+1)}^{(\ell)} - r_{cj}^{(\ell)})\frac{jw(a(\ell)-1)}{e(\ell)} - w\mathbb{I}[j=w]\right) + \mathcal{O}(1/\mathtt{E}) \\
&\triangleq g_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E}).
\end{aligned} \tag{34}$$

Finally, the expected change in the fraction of edges connected to decodable and not decodable GC nodes of degree $j, j = \mathtt{d}, \mathtt{d}+1$, are as follows:

$$\begin{aligned}
&\mathbb{E}\left[\hat{R}_{cj}^{(\ell+1)} - \hat{R}_{cj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}\right] \\
&= P_{p1}\left((p_j\bar{r}_{c(j+1)}^{(\ell)} + \hat{r}_{c(j+1)}^{(\ell)} - \hat{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)}\right) \\
&+ \sum_{w=1}^{j+1} P_{cw}\left((p_j\bar{r}_{c(j+1)}^{(\ell)} + \hat{r}_{c(j+1)}^{(\ell)} - \hat{r}_{cj}^{(\ell)})\frac{jw(a(\ell)-1)}{e(\ell)} - w\mathbb{I}[w=j]\right) + \mathcal{O}(1/\mathtt{E}) \\
&\triangleq \hat{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E})
\end{aligned} \tag{35}$$

$$\begin{aligned}
&\mathbb{E}\left[\bar{R}_{cj}^{(\ell+1)} - \bar{R}_{cj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}\right] \\
&= P_{p1}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\ell)} - \bar{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)}\right) \\
&+ \sum_{w=1}^{j+1} P_{cw}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\ell)} - \bar{r}_{cj}^{(\ell)})\frac{jw(a(\ell)-1)}{e(\ell)} - w\mathbb{I}[w=j]\right) + \mathcal{O}(1/\mathtt{E}) \\
&\triangleq \bar{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E})
\end{aligned} \tag{36}$$

In Appendix A we show that the trend functions in (31)-(36) satisfy Wormald's Theorem requirements. Consequently, the asympotic ($\mathtt{E} \to \infty$) evolution of the process $\mathcal{G}^{(\ell)}/\mathtt{E}$ can be computed by solving the system of differential equations

$$\frac{dl_l(\tau)}{d\tau} = f_i(\mathcal{G}^{(\ell)}/\mathtt{E}), \tag{37}$$

$$\frac{dr_{pj}(\tau)}{d\tau} = g_{pj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \tag{38}$$

$$\frac{dr_{cj}(\tau)}{d\tau} = g_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad j \notin \{\mathtt{d}, \mathtt{d}+1\} \tag{39}$$

$$\frac{d\hat{r}_{cj}(\tau)}{d\tau} = \hat{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad \frac{d\bar{r}_{cj}(\tau)}{d\tau} = \bar{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{40}$$

with initial conditions in (24)-(27). We resort to numerical integration to compute the solution of the system of differential equations for every value of ϵ. Based on this result, a numerical estimate on P-PD threshold for the $\mathcal{C}_{J,K,\nu}$ ensemble is obtained by numerically searching for the highest ϵ value for which the function $r_{p1}^{(\tau)} + \sum_{w=1}^{\mathtt{d}-1} r_{cw}^{(\tau)}/w + \hat{r}_{c\mathtt{d}}^{(\tau)}/\mathtt{d} + \hat{r}_{c(\mathtt{d}+1)}^{(\tau)}/(\mathtt{d}+1)$ remains strictly positive for any $\tau \in [0, \sum_{i=1}^{J} l_i^{(\tau)}/i]$ such that $e(\tau) > 0$.

V. Analysis of the $\mathcal{C}_{J,K,\nu}$ ensemble under P-PD

In this section, we study the asymptotic performance of the $\mathcal{C}_{J,K,\nu}$ ensemble for different base DDs as we vary the fraction ν of GC nodes in the graph. We use high rate base DDs that correspond to regular LDPC code ensembles with variable degree equal to $J = 2$. Further examples with $J > 2$ are discussed in Section VIII. We summarize the parameter of the base DD considered here in Table I. We denote by ϵ_0 the PD threshold of the base LDPC ensemble. Recall that $p_w = 1$ for $w \leq \mathtt{d} - 1$. Each $(p_{\mathtt{d}}, p_{\mathtt{d}+1})$ pair has been determined by finding at least one component code of blocklength K and minimum distance $\mathtt{d}$ that is able to recover a fraction $p_{\mathtt{d}}$ of degree-$\mathtt{d}$ erasure patterns and a fraction $p_{\mathtt{d}+1}$ of degree-$(\mathtt{d}+1)$ erasure patterns. Like this, we ensure that there exists at least one linear block code that satisfies these requierements. We denote this specific block code as the code reference of the family of linear block codes. To find such codes, we performed exhaustive search over the database [23], [24], which implements MAGMA [25] to design block codes with the largest minimum distance. More precisely, for every K, we search for the code with the largest found minimum distance $\mathtt{d}$, and we use the corresponding $p_{\mathtt{d}}$ and $p_{\mathtt{d}+1}$ parameters. These values are listed in Table II and used as a reference for the whole family of linear block codes. The corresponding reference block codes are listed in Appendix B. Note that despite having different blocklength and rate, many reference block codes share $p_{\mathtt{d}}$, $p_{\mathtt{d}+1}$ parameters. We construct $\mathcal{C}_{J,K,\nu}$ ensembles by combining base DDs with the component code families summarized in Table II. For each code ensemble, we compute the P-PD threshold ϵ^* as a function of ν.

TABLE I

BASE DDS, THEIR DESIGN RATES AND ITERATIVE DECODING THRESHOLDS UNDER PD

Base DD	K	$\mathtt{R}_0$	ϵ_0	Gap to capacity $(1-\mathtt{R}_0-\epsilon_0)$
$(2,6)$-regular	6	2/3	0.206	0.127
$(2,7)$-regular	7	5/7	0.167	0.119
$(2,8)$-regular	8	3/4	0.147	0.103
$(2,15)$-regular	15	13/15	0.071	0.062

TABLE II

FAMILIES OF COMPONENT LINEAR BLOCK CODES.

Code Family Index	blocklength K	$\mathtt{d}$	$p_{\mathtt{d}}$	$p_{\mathtt{d}+1}$
I	6	3	80%	0
II	6	4	80%	0
III	7	3	80%	0
IV	7	4	80%	0
V	8	4	80%	0
VI	8	4	91.43%	57.14%
VII	8	5	96.43%	75%
VIII	15	3	92.31%	61.54%
IX	15	4	92.31%	61.54%

A. *Results for* $(2,6)$ *and* $(2,7)$ *base DDs*

Fig. 3 shows the computed P-PD threshold ϵ^* of the $\mathcal{C}_{J,K,\nu}$ ensemble for a base DD $(2,6)$-regular as a function of ν. We consider GC nodes with minimum distance $\mathtt{d}$ equal to 3 and 4 and parameters given by Families I and II in Table II. We also include the BD-PD threshold, which only depends on the minimum distance $\mathtt{d}$ of the component codes and can be computed by solving the system of differential equations in Section IV-B by setting $p_{\mathtt{d}} = p_{\mathtt{d}+1} = 0$. First of all, observe that the P-PD gains in threshold w.r.t. BD-PD is only significant for large values of ν . Further, for either P-PD or BD-PD, using component codes with larger minimum distance ($\mathtt{d} = 4$ instead of $\mathtt{d} = 3$) only pays off for very large values of ν.

Since increasing ν also modifies the code rate $\mathtt{R}(\nu)$ in (4), the comparison in Fig. 3 can be misleading, as we cannot directly evaluate the distance to the channel capacity. In fact, not all values of ν are achievable. We overcome this issue by directly comparing asymptotic threshold and code rate, both defined as parametric curves w.r.t. ν. Denote by $\epsilon^*(\nu)$ the function that gives ϵ^* for a given $\nu \in [0,1]$. Observe in Fig. 3 that $\epsilon^*(\nu)$ is a continuous, strictly increasing function and that for $\nu = 0$ its value

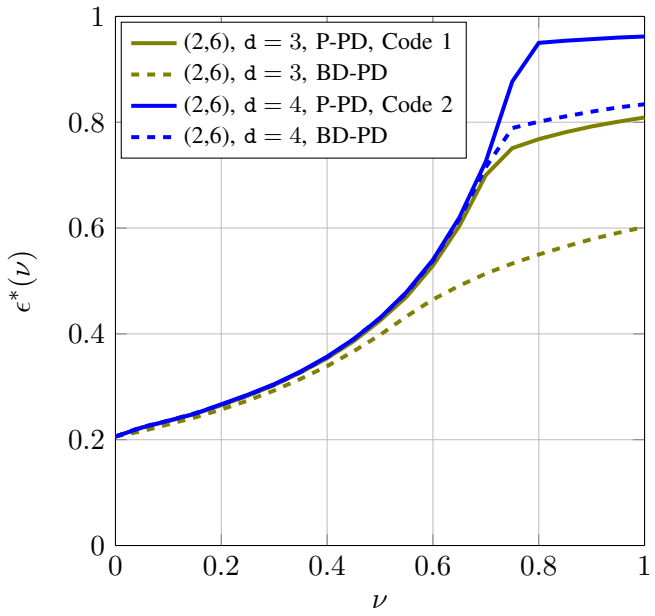
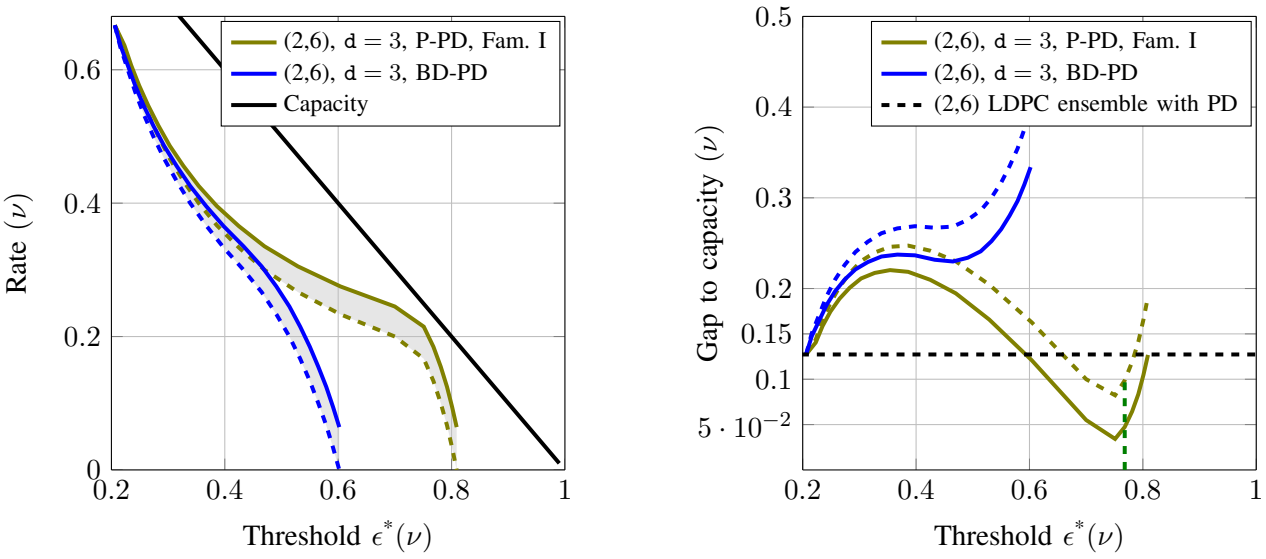


Fig. 3. P-PD and BD-PD thresholds as a function of ν for the $(2, 6)$ base DD.

is equal to ϵ_0, the threshold of the base LDPC ensembles. The inverse of this function, which can be obtained numerically, is denoted by $\nu(\epsilon^*)$ and provides the minimum fraction of GC nodes in the graph required to achieve an ensemble threshold at least ϵ^*. Given the function $\nu(\epsilon^*)$ described above, we use Lemma 2.3 to determine bounds on $\mathtt{R}(\nu)$ for a given targeted decoding threshold ϵ^*. More precisely, by using $\nu(\epsilon^*)$ in (6), we obtain a *converse bound* on the coding rate required to achieve a P-PD decoding threshold equal to ϵ^* using component codes with minimum distance $\mathtt{d}$. Similarly, using $\nu(\epsilon^*)$ in (7), we obtain an *achievable bound* on the coding rate required to achieve a P-PD decoding threshold equal to ϵ^* using linear component codes with minimum distance $\mathtt{d}$. We proceed along the same lines to obtain bounds on the $\mathcal{C}_{J,K,\nu}$ rate for the BD-PD thresholds.

In Fig. 4 (a) we plot these bounds as a function of ϵ^*, both P-PD and BD-PD, using families of block component codes I, with minimum distance $\mathtt{d} = 3$. In Fig. 4 (b), we show the gap to channel capacity computed for each case. In both figures, the value bigger than the point marked by the dashed green line corresponds to the threshold computed at $\hat{\nu}$ in (15). Since $\epsilon^*(\nu)$ is monotonically increasing in ν, any configuration with threshold larger than $\epsilon^*(\hat{\nu})$ has a minimum distance that grows linearly with the block length $\mathtt{n}$. Observe that the performance of both BD-PD and P-PD overlaps for coding rates close to the original rate of the base DD, i.e., for small values of ν.[1] However, while the BD-PD gap to capacity

[1]Recall that by (4), increasing ν yields smaller coding rates.



(a) (b)

Fig. 4. In (a), we plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2, 6)$ and $\mathtt{d} = 3$ component codes as a function of the P-PD and BD-PD thresholds. In (b), we show the gap to channel capacity.

always grows with ν, this is not the case for P-PD. Furthermore, there are values of ν for which the P-PD gap to capacity is smaller than for than the base LDPC ensemble under PD. For the $(2, 6)$ base DD, the P-PD minimum gap to capacity, measured using the achievable rate bound, is 0.0823 for a coding rate of 0.1667. The gap to capacity at $\hat{\nu}$ is from 0.0987 to 0.1269. According to Table I, for this base DD, the PD gap to capacity is 0.1273. Also observe that the region where the $\mathcal{C}_{J,K,\nu}$ ensemble performs better than the base LDPC ensemble is very narrow, and this region does not include the case where all check nodes are GC nodes, i.e., $\nu \to 1$.

Fig. 5 reproduces the results for the code Family II, with minimum distance $\mathtt{d} = 4$. In this case, observe that the P-PD converse bound coincides with the Channel Capacity limit. However, the two bounds are loose and it is uncertain whether we can find an specific block component code in the family that is able to operate close to the converse bound.[2] In fact, the bounds for P-PD and BD-PD overlaps in a large region despite the fact that P-PD using component codes from Family II resolves degree-$\mathtt{d}$ erasure patterns with high probability (0.8).

We observe similar conclusions when we analyze the asymptotic behaviour of the $\mathcal{C}_{J,K,\nu}$ ensemble constructed using a $(2, 7)$ base DD with $\mathtt{d} = 3$ component codes (Fig. 6) and $\mathtt{d} = 4$ component codes. As predicted by Lemma 2.3, when using component codes of blocklength $r = 7$ with minimum distance

[2]Recall that the rate bounds derived in Section II-B only consider the minimum distance of the code family.

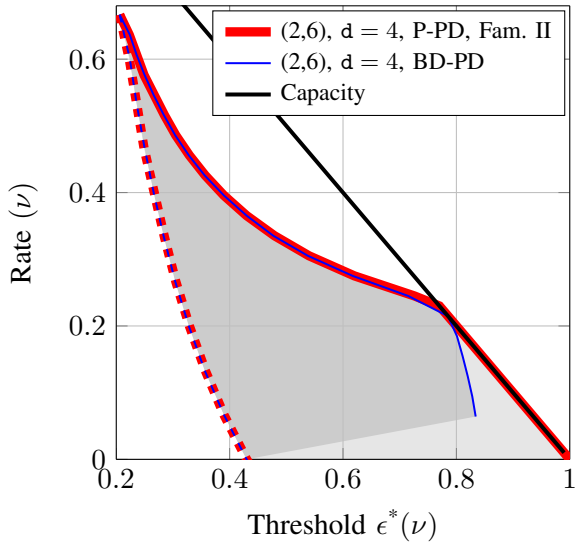


Fig. 5. Bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2, 6)$ and $\mathtt{d} = 4$ component codes as a function of the P-PD and BD-PD thresholds.

$\mathtt{d} = 3$, the converse and achievable bound on the $\mathcal{C}_{J,K,\nu}$ coding rate coincide. And thus, the existence of a linear block component code that satisfies Family III constrains for which the $\mathcal{C}_{J,K,\nu}$ ensemble asymptotically achieves the results in Fig. 6 is guaranteed. Again, there is a region where the P-PD gap to capacity can be reduced with respect to the base LDPC ensemble under PD.

B. Results for higher-density base DDs

We finish this section by extending the above results to base DDs with higher check degree and, thus higher ensemble density. In Fig. 7(a), we consider a $(2, 8)$ base DD with component codes in Code Families V, VI and VII (See Table II). Observe first that rate bounds for Code Families V and VI coincide, even though Code Family VI has better decoding capabilities. In both cases the bounds are loose, but we can still observe a significant improvement w.r.t. component Code Family VII, which has but very large ($\mathtt{d} = 5$) minimum distance, which implies component codes with small coding rate. This again shows that the use of more robust and powerful component codes does not compensate for the quick decay in the GLDPC ensemble coding rate. In Fig. 7(b), we consider a $(2, 15)$ base DDs with component Code Families VIII ($\mathtt{d} = 3$). In this case, the bounds coincide and the gap to capacity is minimized at a coding rate $\mathtt{R} \approx 0.54$ with gap to capacity equal to 0.074, slightly above the gap to capacity for the base LDPC ensemble under PD (0.062). However, at this point the GLDPC ensemble still does not have a minimum distance that grows linearly with the block length.

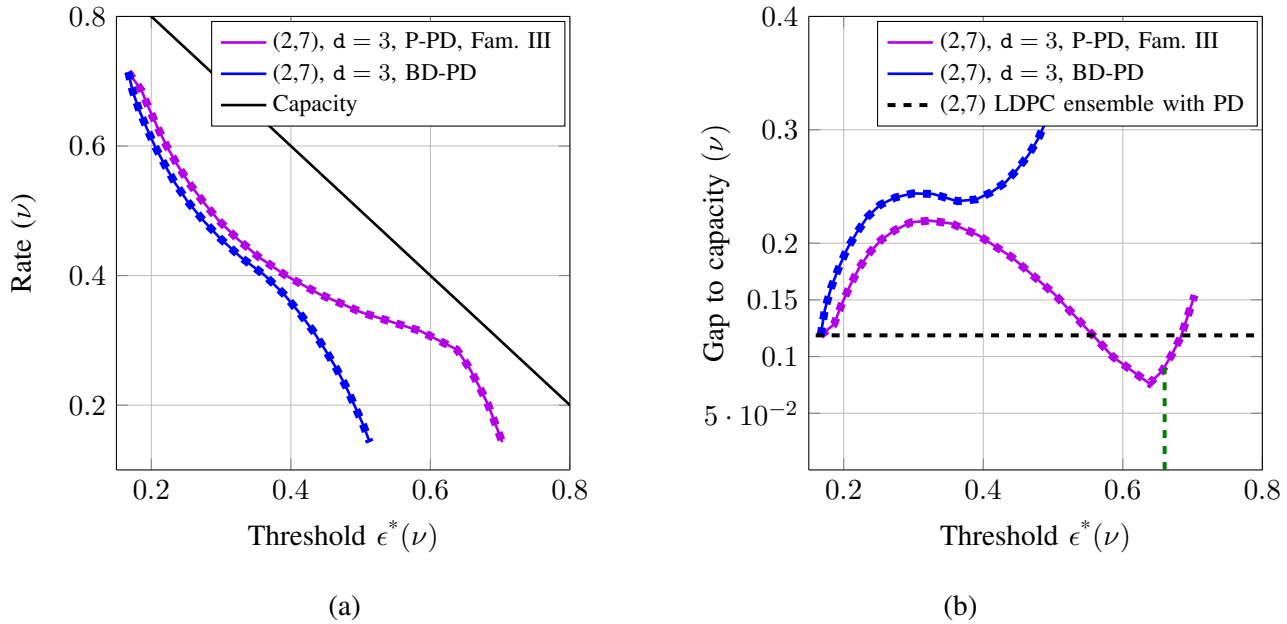


Fig. 6. In (a), we plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2,7)$ as a function of the P-PD and BD-PD thresholds. Note that the bounds overlap in this case. In (b), we show the gap to channel capacity for each case.

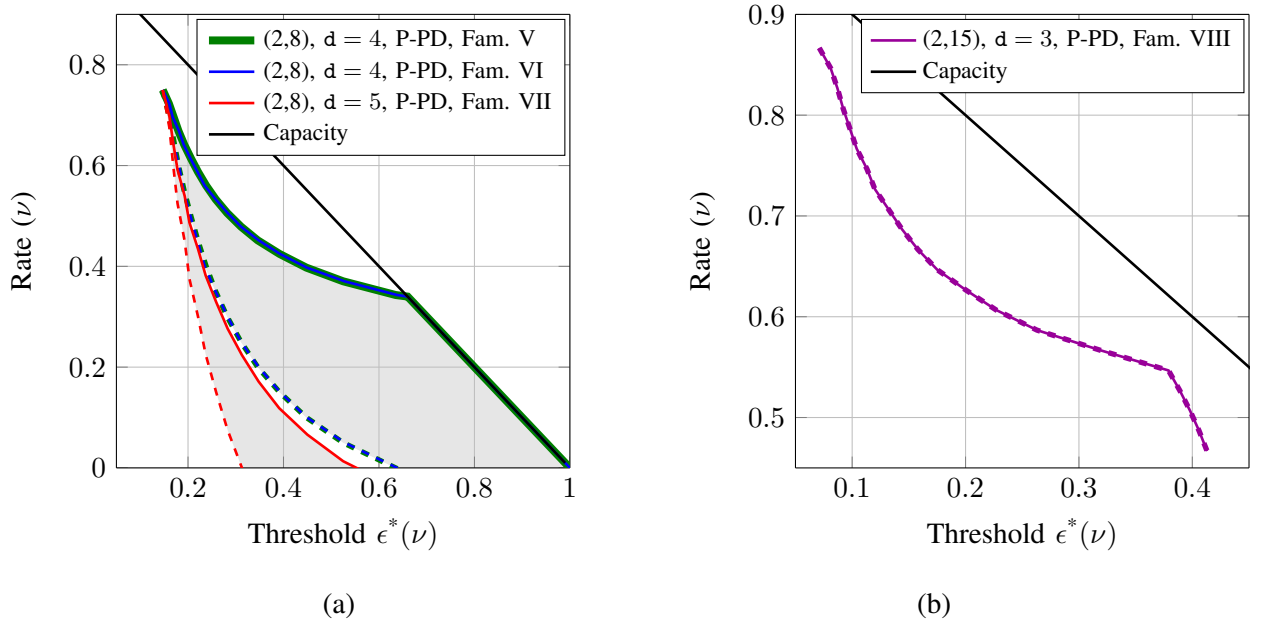


Fig. 7. We plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2,8)$ (a) and $(2,15)$ (b) as a function of the P-PD thresholds.

VI. Selecting specific component codes

By using the bounds on the $\mathcal{C}_{J,K,\nu}$ code rate, we have been able to assess the performance of the $\mathcal{C}_{J,K,\nu}$ ensemble for a whole family of linear block component codes. In certain scenarios the proposed bounds

on the $\mathcal{C}_{J,K,\nu}$ code rate provide meaningful design information about the asymptotic behavior of the ensemble. The natural question that arises at this point is whether we can find specific component codes within the family that improve the achievable bound in (7), reducing the gap to the rate converse bound in (6). In this section, we analyze the asymptotic performance of $\mathcal{C}_{J,K,\nu}$ when component codes are chosen from the the list of reference linear block component codes summarized in Table III. The construction of these linear block codes is detailed in [23], and their generator matrix is given in Appendix B.

TABLE III

SUMMARY OF LINEAR BLOCK COMPONENT CODES IN APPENDIX B. $\mathtt{k}$ IS THE NUMBER OF ROWS IN THE CODE PARITY-CHECK MATRIX.

Code Index	blocklength K	$\mathtt{k}$	Rate	Code Family it Belongs
A	6	3	1/2	I
B	6	4	1/3	II
C	7	3	4/7	III
D	7	4	3/7	IV
E	8	4	1/2	V
F	8	5	3/8	VI
G	8	6	1/4	VII
H	15	4	11/15	VIII
I	15	5	2/3	IX

Once we fix a particular class of component codes to be used at GC nodes, we can replace the $\mathcal{C}_{J,K,\nu}$ code bounds by the actual code rate in (4). In Fig. 8 we plot the $\mathcal{C}_{J,K,\nu}$ coding rate and the achievable bound of the corresponding family of codes for $(2,6)$-regular and $(2,7)$-regular base DDs. Results for $(2,8)$-regular and $(2,15)$-regular base DDs can be found in Fig. 9. Observe that, with the component codes proposed, we are able to perform at least as good as the achievable bound of the corresponding family of block component codes. In some cases, the achievable bound is significantly improved. We highlight those points for which, asymptotically, the $\mathcal{C}_{J,K,\nu}$ ensemble with the proposed linear block component codes under P-PD operates closer to channel capacity than the base LDPC code ensemble under PD. As already discussed in Section V, observe that, for every base DD, the use of component codes with higher minimum distance does not improve the GLDPC asymptotic performance.

A. Growth Rate of the Weight Distribution

Upon selecting a specific block code, we can compute the weight spectral shape $G(\alpha)$ in (14) using the tools proposed in [10]. In Fig. 10, we plot $G(\alpha)$ for the $\mathcal{C}_{J,K,\nu}$ ensembles and different values of ν,

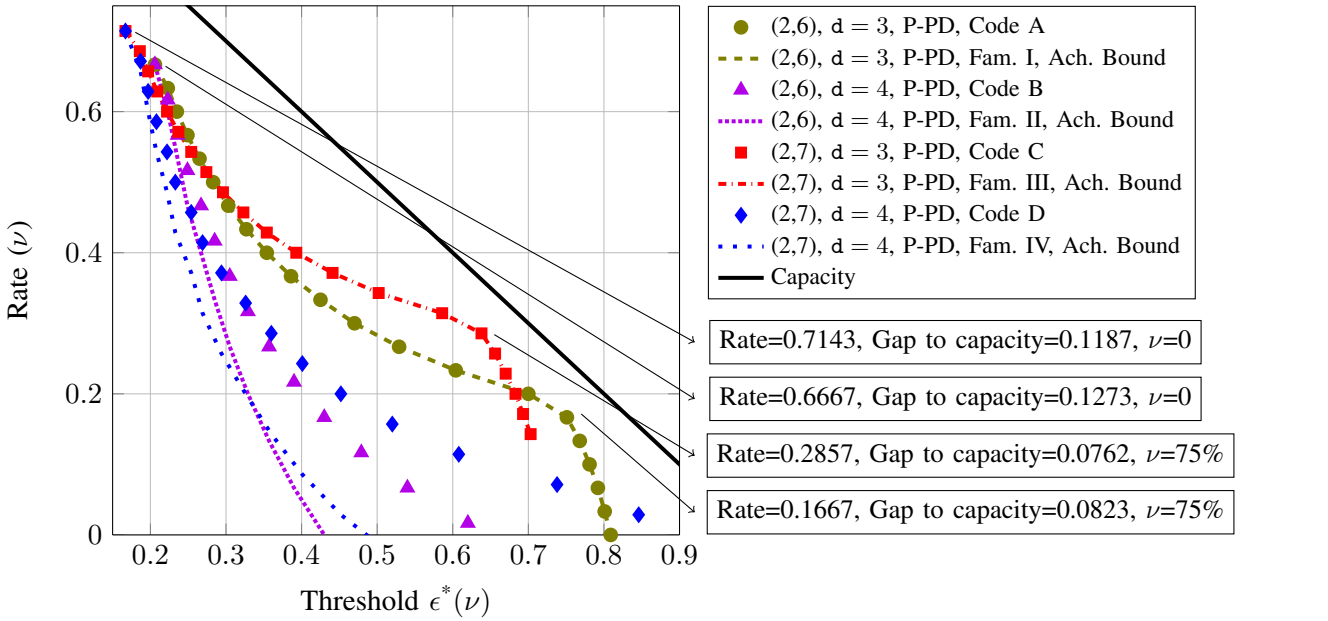


Fig. 8. $\mathcal{C}_{J,K,\nu}$ coding rate and achievable bound in (7) for $(2,6)$ and $(2,7)$ base DDs and component codes from Table III as a function of the P-PD decoding threshold.

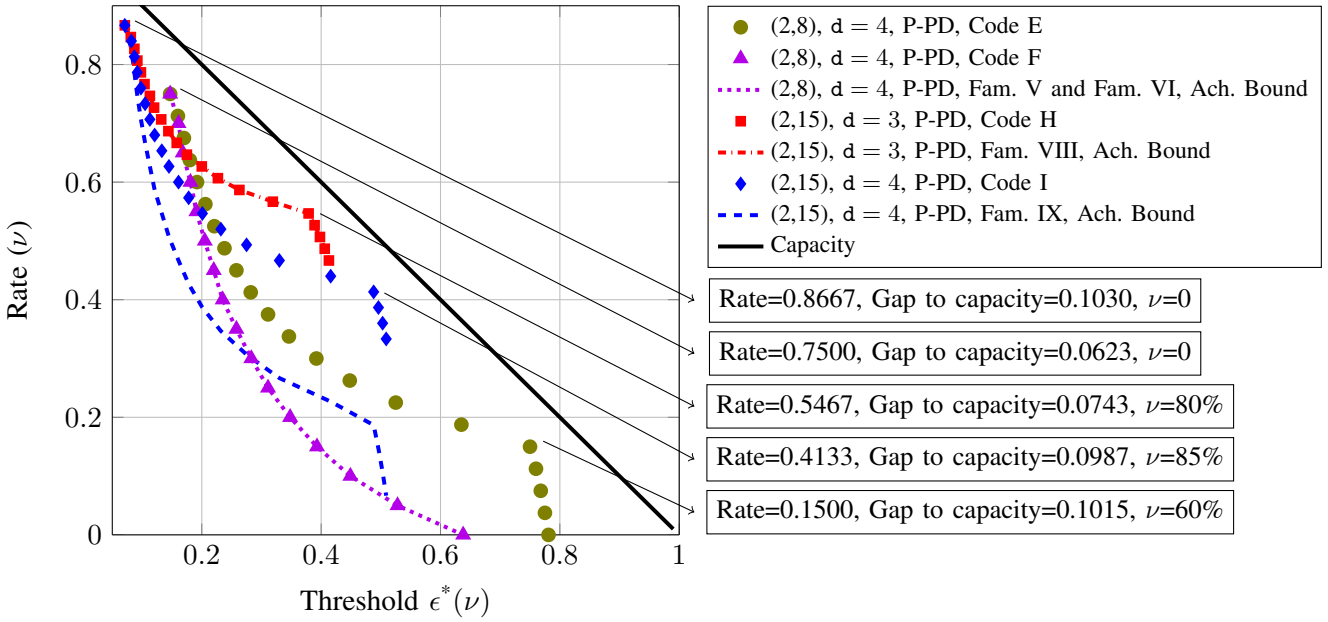


Fig. 9. Actual $\mathcal{C}_{J,K,\nu}$ coding rate and achievable bound in (7) under the P-PD approximation to the ML-PD threshold for different base DDs and component codes.

computed for the $(2,6)$-regular base DD with Code A as component code (a) and the $(2,7)$-regular base DD with Code C as component code (See Table III). Recall that the critical exponent codeword weight ratio is defined as $\hat{\alpha} \triangleq \inf\{\alpha \geq 0 | G(\alpha) \geq 0\}$. $\hat{\alpha}$ is highlighted in the plots by a thick dot in each curve. Observe that, as predicted, $\hat{\alpha} = 0$ at $\nu = \hat{\nu}$, given in (15). As we make ν larger, $\hat{\alpha}$ grows and it achieves its maximum at $\nu = 1$. The results in Fig. 8 along with those in Fig. 10 indicate that there is a trade-off between $\epsilon^*(\nu)$, the threshold of the code, and $\hat{\alpha}(\nu)$, the critical exponent codeword weight ratio. As an example, we include values of both quantities in Table IV for the $(2,6)$-regular base DD with Code A as component code .

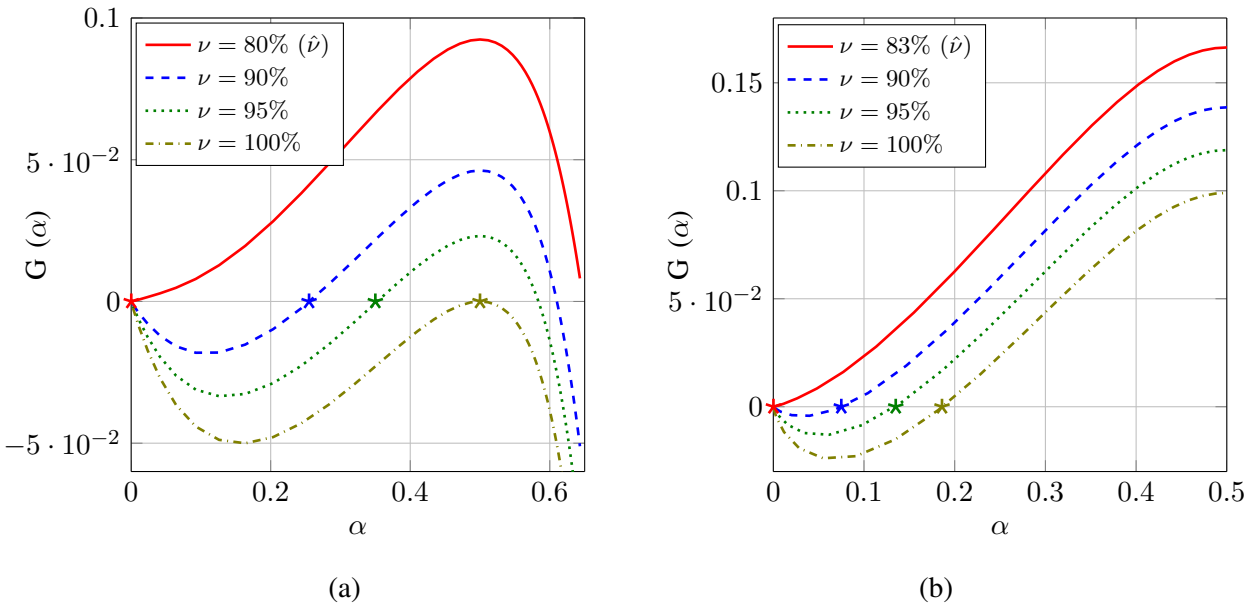


Fig. 10. In (a) we plot the weight spectral shape $G(\alpha)$ in (14) of the $\mathcal{C}_{J,K,\nu}$ ensemble based on $(2,6)$-regular base DD with Code A component codes. In (b), we reproduce the results for the $\mathcal{C}_{J,K,\nu}$ ensemble based on a $(2,7)$-regular base DD and Code B component codes (b).

VII. Using random puncturing to reduce the gap to capacity

We have proposed the P-PD algorithm as a flexible model to analyze beyond-BD decoding algorithm at GC nodes. Also, the evaluation of both coding rate and of iterative decoding threshold are decoupled problems. This provides a flexible analysis framework where we could explore additional techniques to modify the designs presented above and further reduce gap to capacity. In this section and the following one, we consider two relevant examples. Here, we consider the use of random puncturing to easily accommodate the coding rate by dropping the transmission of a certain fraction of coded bits [17]. In

TABLE IV

$\hat{\alpha}$, ϵ^* AND GAP TO CAPACITY FOR DIFFERENT VALUES OF ν, COMPUTED FOR THE $(2,6)$-BASE DD WITH CODE A COMPONENT CODES

ν	α	ϵ^*	Gap to capacity
80%	0	0.768	0.0987
87.5%	0.2049	0.788	0.1287
90%	0.2556	0.792	0.1413
92.5%	0.3038	0.797	0.1530
95%	0.3526	0.801	0.1657
97.5%	0.4056	0.806	0.1773
100%	0.6078	0.809	0.1910

the next section, a simple model of doubly-generalized LDPC (DG-LDPC) code ensemble is analyzed [18], [19], [5].

As illustrated in [17], a linear code is *punctured* by removing a set of columns from its generator matrix. After puncturing at random a fraction ξ of the coded bits in the $\mathcal{C}_{J,K,\nu}$ ensemble, of coding rate $\mathtt{R}(\nu)$, the resulting coding rate is

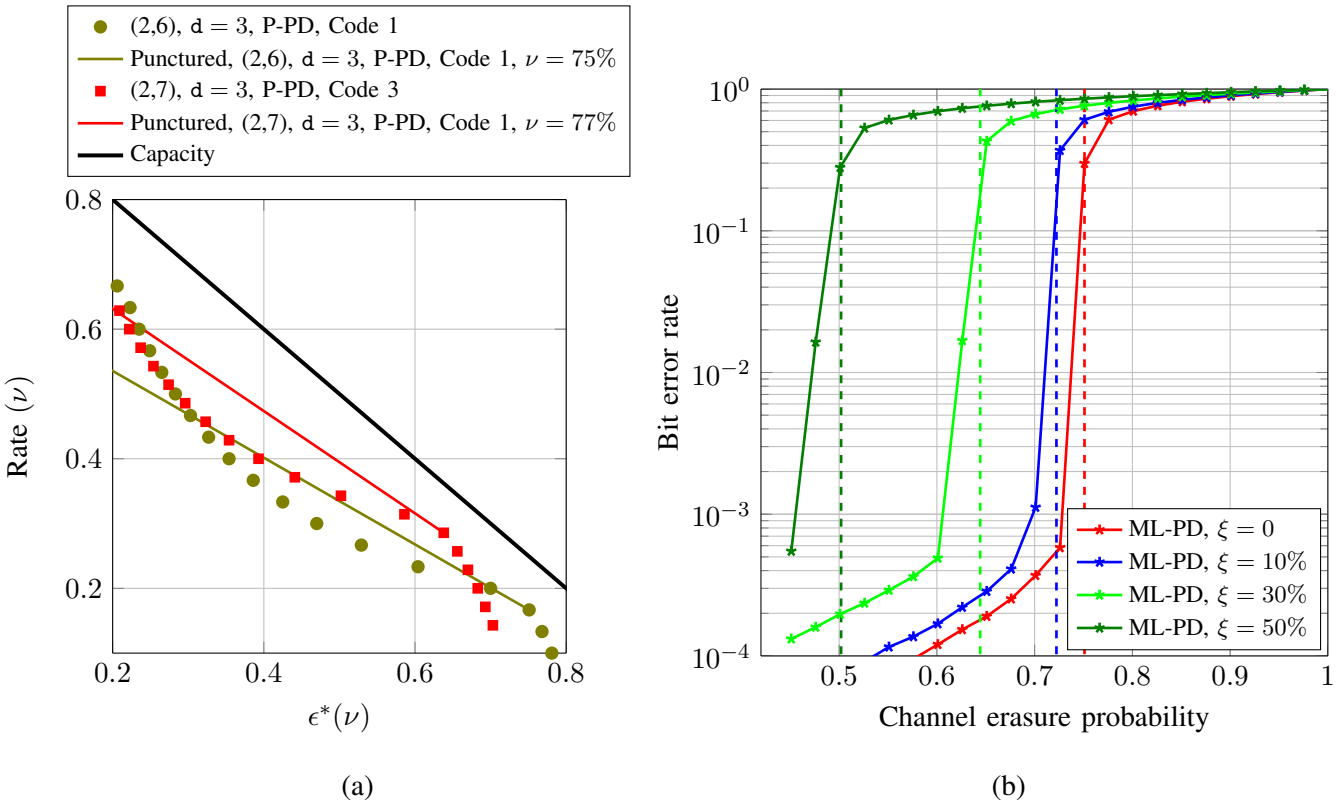
$$\mathtt{R}(\nu,\xi) = \frac{\mathtt{R}(\nu)}{1-\xi}, \quad \xi \in [0,1). \tag{41}$$

In [17] the authors derive a simple analytic expression for the iterative belief propagation (BP) decoding threshold of a randomly punctured LDPC code ensemble on the binary erasure channel (BEC). Following their proof, it is straightforward to check that the same results apply to a randomly punctured GLDPC code ensemble. The result reads as follows. Given the $\mathcal{C}_{J,K,\nu}$ ensemble with iterative decoding threshold $\epsilon^*(\nu)$, then the threshold $\epsilon^*(\nu,\xi)$ of the GLDPC ensemble that results after randomly puncture a fraction ξ of the coded bits is related to the unpunctured case as follows:

$$\epsilon^*(\nu,\xi) = 1 - \frac{1-\epsilon^*(\nu)}{1-\xi}, \tag{42}$$

where note that the larger the unpunctured threshold $\epsilon^*(\nu)$ is, the larger the threshold of the punctured ensemble will be. In this regard, we can think of the design of a punctured GLDPC ensemble as a two stage process: first, the GLDPC code ensemble can be designed by choosing ν to minimize the gap to capacity. Fixed ν, then by combining (41) and (42), we can analyze the overall gap to capacity as we increasing the code rate. We perform this experiment in Fig. 11 (a) for the $(2,6)$ and the $(2,7)$ base DDs and component codes A and C respectively. With markers we show the $\mathcal{C}_{J,K,\nu}$ threshold-rate curve as we increase the fraction of GC nodes in the graph. Solid lines indicate the evolution of the rate and

threshold of the punctured ensemble for fixed ν as increase the puncturing fraction ξ. Observe that with puncturing it is possible to increase the coding rate and obtain an iterative decoding threshold that is closer to capacity than those obtained by the $\mathcal{C}_{J,K,\nu}$ ensemble. The accuracy of the predicted threshold can be observed in Fig. 11 (b), where we include both the simulated P-PD performance for the $(2,6)$ base DD with component code A, $\mathtt{n} = 10000$ bits, and different values of the puncturing rate ξ, and the predicted iterative threshold included using vertical dashed lines.



(a) (b)

Fig. 11. In (a), with markers we show the $\mathcal{C}_{J,K,\nu}$ threshold-rate curve for the $(2,6)$ and the $(2,7)$ base DDs and component codes A and C respectively. Solid lines indicate the evolution of the rate and threshold of the puncture ensemble for fixed ν as increase the puncturing fraction ξ. In (b), we show the simulated P-PD performance for the $(2,6)$ base DD with component code A, $\mathtt{n} = 10000$ bits, and different values of the puncturing rate ξ. Vertical dashed lines indicate the predicted iterative threshold.

VIII. Including generalized variable nodes to reduce the gap to capacity

A different technique that can potentially help to find a better balance between coding rate and threshold is the inclusion of generalized variable nodes, giving rise to a doubly-generalized LDPC code ensemble [18]. In this section we develop an example with a simple class of DG-LDPC ensemble. We modify the $\mathcal{C}_{J,K,\nu}$ ensemble by replacing a certain fraction β of regular variable (RV) nodes by generalized variable (GV) nodes, see Fig. 12 (a). Degree-J RV nodes in the $\mathcal{C}_{J,K,\nu}$ graph can be seen as rate $1/J$ repetition

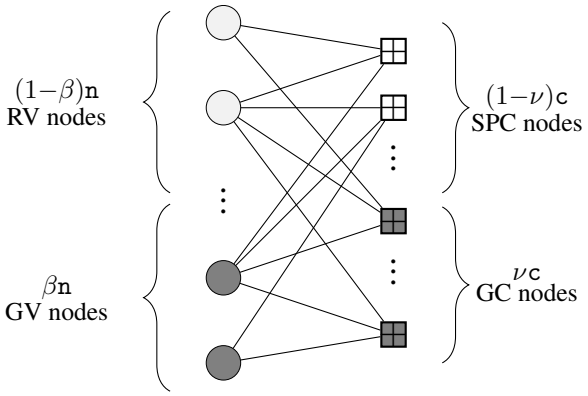


Fig. 12. Tanner graph of the DG-LDPC code ensemble.

nodes of block length J, where the input to the repetition code represents one bit of the DG-DLPC codeword. On the other hand, degree-J GV nodes are characterized by a $(J, \mathtt{m})$ linear block code, where the input to the variable component code represents $\mathtt{m}$ bits of the DG-LDPC codeword. Thus, the total block length of the DG-LDPC code ensemble is $\mathtt{n}' = (1-\beta)\mathtt{n} + \beta\mathtt{nm}$, where $\mathtt{n}$ is the number of variable nodes in the graph, both RV and GV nodes. In the following, we will assume $J = 3$ and the following generator matrix for GV nodes:

$$\mathrm{G} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{43}$$

Thus, each GV node encodes two bits of the DG-LDPC codeword. Denote the ensemble by $\mathcal{C}_{3,K,\nu,\beta}$. If component codes at GC nodes are characterized by linear block code with a $\mathtt{k}$-row parity check matrix, an easy calculation shows that the coding rate of the ensemble is:

$$\mathtt{R}(\alpha, \beta) = 1 - (1 - \mathtt{R}_0)\left(\frac{1 + (\mathtt{k}-1)\nu}{1+\beta}\right). \tag{44}$$

Also, we assume a given triple $(\mathtt{d}, p_{\mathtt{d}}, p_{\mathtt{d}+1})$ that characterizes component GC nodes. On the other hand, note the code associated with the generator matrix in (43) has minimum distance 2 and it can only decode erasure patterns of weight one.

A. Decoding via P-PD

As before, RV nodes can be removed from the graph with probability $1-\epsilon$. Regarding GV nodes, we have to consider the following scenarios:

- With probability $(1-\epsilon)^2$ the two coded bits are correctly received and the GV node can be removed from the graph.

- With probability $2\epsilon(1-\epsilon)$, one of the two coded bits is received. In this case, we can replace the GV node in the graph by a degree-2 RV node.
- With probability ϵ^2 the GV node remains in the graph as a degree-3 node. Since code spanned by (43) can only decode one error, during P-PD the GV node needs to lose at least two edges before it can be removed from the graph. Also, once it loses one edge, it can be replaced by a degree-2 RV node.

Decoding will be performed via P-PD. A small modification is only required at step 2) in the P-PD Algorithm in Section III by only removing from the graph GV nodes with degree less than 3.

B. Degree Distribution and Asymptotic Analysis

While no change is needed to describe the DD of the residual DG-LDPC code ensemble during P-PD, see Section IV-B, additional definitions at the variable side are needed to tackle both RV nodes and GV nodes. Let $L_{r2}^{(\ell)}$ and $L_{r3}^{(\ell)}$ represents the total number of edges in the graph connected to RV nodes of degree 2 and 3 respectively, after the ℓ-iteration of the decoder. Also, let $L_{g3}^{(\ell)}$ be the total number of edges in the graph connected to GV nodes of degree 3. Also we define,

$$l_{r2}^{(\ell)} = \frac{L_{r2}^{(\ell)}}{\mathtt{E}},\ l_{r3}^{(\ell)} = \frac{L_{r3}^{(\ell)}}{\mathtt{E}},\ l_{g3}^{(\ell)} = \frac{L_{g3}^{(\ell)}}{\mathtt{E}}. \tag{45}$$

Given these definitions, the expected left DD of the DG-LDPC graph after P-PD initialization is:

$$\mathbb{E}[l_{r3}^{(0)}] = \epsilon(1-\beta), \tag{46}$$

$$\mathbb{E}[l_{r2}^{(0)}] = \frac{2(2\epsilon(1-\epsilon))\beta\mathtt{n}}{3\mathtt{n}} = 4\beta\epsilon(1-\epsilon)/3, \tag{47}$$

$$\mathbb{E}[l_{g3}^{(0)}] = \epsilon^2\beta \tag{48}$$

Regarding the expected DD of the check nodes after P-PD initialization, note that, in average, the probability of an edge not to be removed from the graph is

$$\epsilon' = \frac{(\epsilon(1-\beta) + 4\beta\epsilon(1-\epsilon)/3 + \epsilon^2\beta)3\mathtt{n}}{3\mathtt{n}} = \epsilon(1+\beta(1-\epsilon)/3), \tag{49}$$

and hence we can use the initialization expressions in (25) and (27) by replacing ϵ by ϵ'. Finally, in order to obtain the system of differential equations that encodes the asymptotic evolution of the graph, we need to derive the expected evolution of the quantities in (45) after one step of the P-PD, i.e., the

corresponding the trend functions. On the left side, given the rules update rules for GV nodes described in Section VIII-A we have

$$\mathbb{E}\left[L_{r3}^{(\ell+1)} - L_{r3}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = -\frac{3l_{r3}^{(\ell)}}{e(\ell)}\left(P_{p1} + \sum_{w=1}^{\mathtt{d}+1} wP_{cw}\right) + \mathcal{O}(1/\mathtt{E}), \tag{50}$$

$$\mathbb{E}\left[L_{r2}^{(\ell+1)} - L_{r2}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = \left(\frac{2l_{g3}^{(\ell)}}{e(\ell)} - \frac{2l_{r2}^{(\ell)}}{e(\ell)}\right)\left(P_{p1} + \sum_{w=1}^{\mathtt{d}+1} wP_{cw}\right) + \mathcal{O}(1/\mathtt{E}), \tag{51}$$

$$\mathbb{E}\left[L_{g3}^{(\ell+1)} - L_{g3}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = -\frac{3l_{g3}^{(\ell)}}{e(\ell)}\left(P_{p1} + \sum_{w=1}^{\mathtt{d}+1} wP_{cw}\right) + \mathcal{O}(1/\mathtt{E}). \tag{52}$$

On the right side, the trend functions determined by (33)-(36) remain unchanged. Note that now the average left degree is computed as $a(\ell) = (3l_{r3}^{(\ell)} + 2l_{r2}^{(\ell)} + l_{g3}^{(\ell)})/e(\ell)$. Finally, the iterative decoding threshold of the $\mathcal{C}_{3,K,\nu,\beta}$ code ensemble can be estimated after numerical integration of the system of differential equations induced by the trend functions in (33)-(36) and (50)-(52).

C. Results for the $(3, 6)$ and $(3, 7)$ base DDs

Fig. 13 shows the computed rate-threshold curve as ν for both the $\mathcal{C}_{3,K,\nu,\beta}$ with $\beta = 0$, i.e., when the code graph has no generalized variable nodes, and with $\beta = 0.3$. We use a $(3, 6)$-base DD and code A in Table III as generalized component code as GC nodes. While in the first case the minimun gap to capacity is achieved for the base LDPC code ensemble (a gap of 0.0710), observe that by using a certain amount of generalized variable nodes we are able to reduce this gap to 0.0592, so far, it is the smallest gap to capacity we have found. Furthermore, since all variable nodes in the graph have degree 3, by lemma 2.3, we know that the code ensemble has a minimum distance that grows linearly with the block length. Fig. 14 reproduces the results with similar conclusions for a $(3, 7)$ base DDs with code C as generalized component code as GC nodes.

IX. Conclusions and Future Work

One of the main contributions of the presented paper is the methodology followed itself. By introducing a certain amount of GC nodes in the LDPC Tanner graph, we analyze if the gap to channel capacity of this designed $\mathcal{C}_{J,K,\nu}$ ensemble is reduced at the resulting coding rate. We have proposed the P-PD algorithm as a flexible model to analyze beyond-BD decoding algorithm at GC nodes. The P-PD predicts accurately ML decoding at GC nodes and allows mathematical analysis of the code performance. Note that in our analysis, the evaluation of both coding rate and of iterative decoding threshold are decoupled problems. In this regard, broader classes of component codes or improved decoding methods at GC nodes could be incorporated in a systematic way. The growth rate of the weight distribution of the example

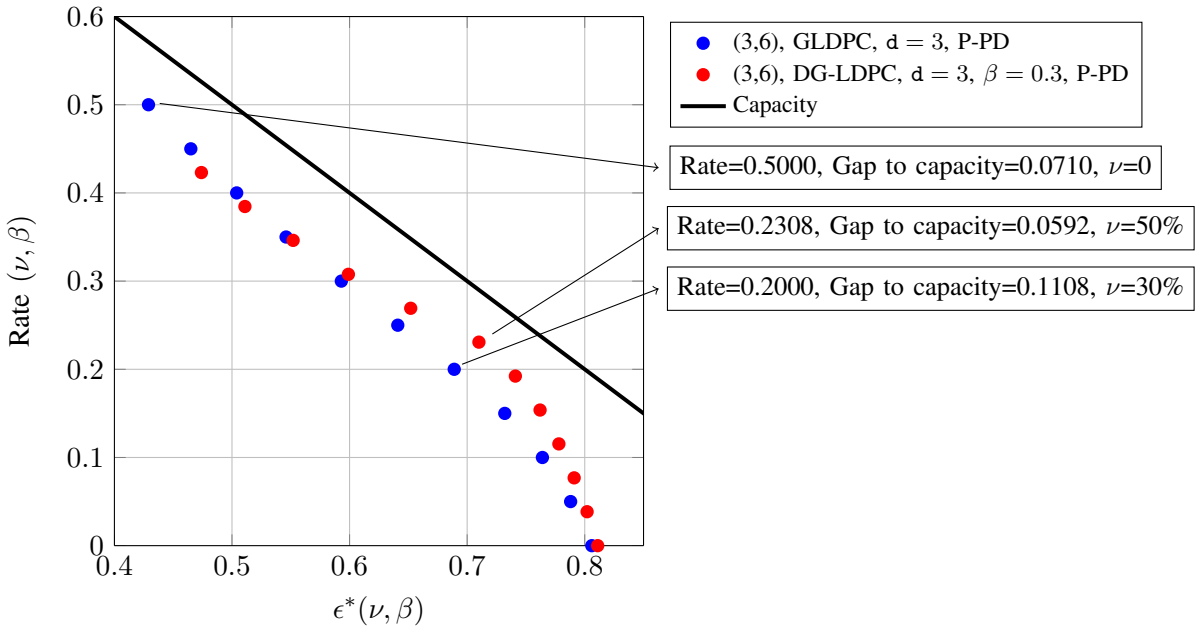


Fig. 13. $\mathcal{C}_{\overline{\lambda},K,\nu,\beta}$ coding rate as a function of P-PD approximation to the ML-PD threshold for (3,6)-regular base DD, Code A component code, G_1 generator matrix of GV nodes and $\beta = 0.3$.

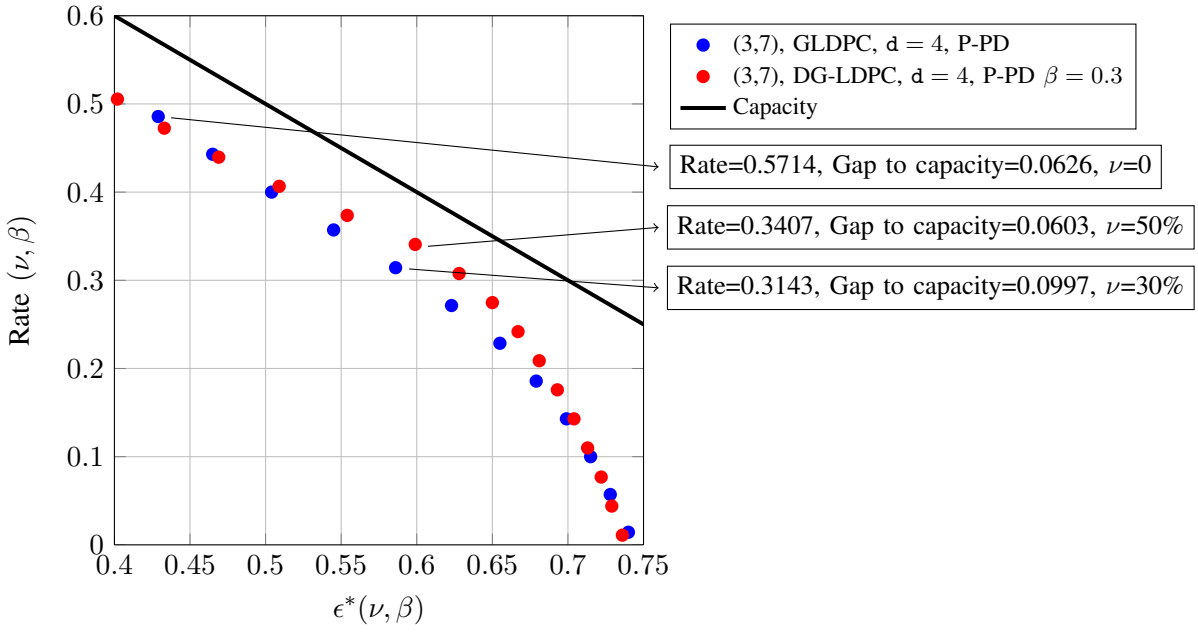


Fig. 14. $\mathcal{C}_{\overline{\lambda},K,\nu,\beta}$ coding rate as a function of P-PD approximation to the ML-PD threshold for (3,7)-regular base DD, Code C component code, G_1 generator matrix of GV nodes and $\beta = 0.3$.

GLDPC code ensembles is analyzed, then we give the trade-off between the threshold of the code, the gap to capacity and the critical exponent codeword weight ratio. Finally, we apply random puncturing and D-GLDPC based on the designed GLDPC code ensembles to explore additional techniques to modify the designed GLDPC codes and future reduce the gap to capacity. The technique proposed can be used to explore designable GLDPC codes and can be extended in a broader area, such as spatially coupled LDPC codes.

APPENDIX A
DERIVATION OF THE EXPECTATIONS IN (32)-(36)

In this section we prove that the evolution of $\{\mathcal{G}^{(\ell)}, \ell \in \mathbb{N}_+\}$ under P-PD satisfies the three conditions of Wormald's theorem so that we can derive a system of differential equations to predict its asymptotic behavior. We start by computing the conditional expected evolution of all elements in the residual DD $\mathcal{G}^{(\ell)}$ after one P-PD iteration.

A. Evolution of left edge degrees in the Tanner graph after one P-PD iteration

Assume the residual graph at time ℓ is observed. Our aim is to evaluate

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right], \tag{53}$$

for $i = 1, 2, ..., J$. Given the graph DD $\mathcal{G}^{(\ell)}$, recall that P_{p1} represents the probability of P-PD selecting a degree-one SPC node in the current iteration, and P_{cj} the probability of selecting a degree-j decodable GC node, $j = 1, \ldots, \mathtt{d}+1$. We can decompose the above expectation by averaging among each possible type of check node chosen to be removed, namely,

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = P_{p1}\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] + \sum_{w=1}^{\mathtt{d}+1} P_{cw}\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right],$$

where Deg_{p1} indicates that the BP-DP decoder removes a degree-one SPC node from the graph, and Deg_{cw} indicates that the BP-DP decoder removes a degree-w decodable GC node. Computing the expectation in the first case, i.e. Deg_{p1}, is similar to that derived for PD with LDPC ensembles in [11]. We include its derivation for completeness. The probability that the edge adjacent to the removed degree-one SPC node has left degree i is $l_i(\ell)/e(\ell)$. In such a case, after deleting this variable node the graph loses $i-1$ additional edges adjacent to this variable node and, hence

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = -\frac{il_i^{(\ell)}}{e(\ell)}. \tag{54}$$

When the P-PD decoder removes a decodable degree-w GC node, this node is connected to w variable nodes that are also removed from the residual Tanner graph, along with their connected edges. The

probability that the first of the w edges is connected to degree-i variable nodes is $\frac{\mathtt{E}l_i^{(\ell)}/e(\ell)}{\mathtt{E}} = l_i^{(\ell)}/e(\ell)$, the second edge of the w edges is connected to degree-i variable nodes is $\frac{\mathtt{E}l_i^{(\ell)}/e(\ell)-1}{\mathtt{E}} = l_i^{(\ell)}/e(\ell) - \mathcal{O}(1/\mathtt{E})$, where $\mathcal{O}(1/\mathtt{E})$ term accounts for the fact that we are computing the joint probability of the left degree of w edges in the graph, with w small and independent of $\mathtt{E}$. For instance, assume we have a graph with N variable nodes, half of them have degree 2 and half of them have degree 3. E.g., $\mathtt{E} = \frac{N}{2}2 + \frac{N}{2}3 = \frac{5}{2}N$. Assume we select at random 3 edges to remove from the graph and denote by X_2 the R. V. that gives how many variable nodes of left degree 2 we remove. An easy calculation shows that

$$\begin{aligned}
\mathbb{P}\left[X_2 = 0\right] &= \left(\frac{\frac{3}{2}N}{\frac{5}{2}N}\right)\left(\frac{\frac{3}{2}N-1}{\frac{5}{2}N-1}\right)\left(\frac{\frac{3}{2}N-2}{\frac{5}{2}N-2}\right),\\
\mathbb{P}\left[X_2 = 1\right] &= 3\left(\frac{N}{\frac{5}{2}N}\right)\left(\frac{\frac{3}{2}N-1}{\frac{5}{2}N-1}\right)\left(\frac{\frac{3}{2}N}{\frac{5}{2}N-2}\right),\\
\mathbb{P}\left[X_2 = 2\right] &= 3\left(\frac{N}{\frac{5}{2}N}\right)\left(\frac{N-1}{\frac{5}{2}N-1}\right)\left(\frac{\frac{3}{2}N}{\frac{5}{2}N-2}\right),\\
\mathbb{P}\left[X_2 = 3\right] &= \left(\frac{N}{\frac{5}{2}N}\right)\left(\frac{N-1}{\frac{5}{2}N-1}\right)\left(\frac{N-2}{\frac{5}{2}N-2}\right).
\end{aligned}$$

Observe that, since $\mathtt{E} \propto N$

$$\begin{aligned}
\mathbb{P}\left[X_2 = 0\right] &= \left(\frac{3/2}{5/2}\right)^3 - \mathcal{O}(1/\mathtt{E}) - \mathcal{O}(1/\mathtt{E}^2)\\
\mathbb{P}\left[X_2 = 1\right] &= 3\frac{1}{5/2}\left(\frac{3/2}{5/2}\right)^2 - \mathcal{O}(1/\mathtt{E}) - \mathcal{O}(1/\mathtt{E}^2)\\
\mathbb{P}\left[X_2 = 2\right] &= 3\left(\frac{1}{5/2}\right)^2\frac{3/2}{5/2} - \mathcal{O}(1/\mathtt{E}) - \mathcal{O}(1/\mathtt{E}^2)\\
\mathbb{P}\left[X_2 = 3\right] &= \left(\frac{1}{5/2}\right)^3 - \mathcal{O}(1/\mathtt{E}) - \mathcal{O}(1/\mathtt{E}^2)
\end{aligned}$$

Thus, in the limit $\mathtt{E} \to \infty$ we have

$$X_2 \sim \mathrm{B}\left(3, \frac{1}{5/2}\right) \Rightarrow \mathbb{E}\left[X_2\right] = 3\frac{1}{5/2}$$

And hence, in terms of edges of the left degree 2 lost is

$$\mathbb{E}\left[L_2^{(\ell+1)} - L_2^{(\ell)}\right] = -3\frac{2}{5/2},$$

which minimize (54). Consequently, the probability in the limit, that all of the w edges are connected to a degree-i variable nodes is $wl_i^{(\ell)}/e(\ell)$, thus

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)}\middle|\mathcal{G}^{(\ell)}\right] = -\frac{iwl_i^{(\ell)}}{e(\ell)} + \mathcal{O}(1/\mathtt{E}).$$

Consequently we have,

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = -\frac{il_i^{(\ell)}}{e(\ell)}\left(P_{p1} + \sum_{w=1}^{\mathtt{d}+1} wP_{cw}\right) + \mathcal{O}(1/\mathtt{E}) \doteq f_i(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E}). \tag{55}$$

Note that $f_i(\mathcal{G}^{(\ell)}/\mathtt{E})$ depends on every component in DD in $\mathcal{G}^{(\ell)}$, normalized by $\mathtt{E}$. Furthermore, (55) is of the form required by Condition 2) of Wormald's theorem. Also note that $|L_i^{(\ell+1)} - L_i^{(\ell)}| \leq x(\mathtt{d}+1)$ for any $i = 1, \ldots, J$ and any iteration, so Condition 1) in Wormald's theorem is also satisfied. Proving that $f_i(\cdot)$ in (55) is Lipschitz continuous follows the exact same argument as in [14].

B. Evolution of edge degrees on the right side of GLDPC graph

Assume that the residual graph at time ℓ is observed. Our goal now is to evaluate

$$\begin{aligned}
&\mathbb{E}\left[R_{pj}^{(\ell+1)} - R_{pj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right], \quad j = 1, \ldots, K,\\
&\mathbb{E}\left[R_{cj}^{(\ell+1)} - R_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right], \quad j = 1, \ldots, K \text{ and } j \notin \{\mathtt{d}, \mathtt{d}+1\}\\
&\mathbb{E}\left[\hat{R}_{cj}^{(\ell+1)} - \hat{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right], \quad j \in \{\mathtt{d}, \mathtt{d}+1\}\\
&\mathbb{E}\left[\bar{R}_{cj}^{(\ell+1)} - \bar{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right], \quad j \in \{\mathtt{d}, \mathtt{d}+1\}
\end{aligned}$$

As before, computing these terms is done by conditioning on the type of check node removed at the current P-PD iteration. When the decoder removes a degree-one SPC node (Deg_{p1} case), an edge with right SPC degree one is removed together with the variable node connected to it. The average degree of this variable node is $a(\ell) = \sum il_i^{(\ell)}/e(\ell)$. If one of the $a(\ell) - 1$ additional edges to be removed has right SPC degree j, after deleting such edges the graph loses j edges with right SPC degree j and gains $j-1$ edges with right SPC degree $j-1$. For the component R_{p1} we should take into account the removed degree-one SPC, and thus

$$\mathbb{E}\left[R_{p1}^{(\ell+1)} - R_{p1}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = (r_{p2}^{(\ell)} - r_{p1}^{(\ell)})\frac{(a(\ell)-1)}{e(\ell)} - 1 + \mathcal{O}(1/\mathtt{E}). \tag{56}$$

A similar argument holds for every possible right degree (SPC or GN). Following this idea, we obtain

$$\begin{aligned}
&\mathbb{E}\left[R_{pj}^{(\ell+1)} - R_{pj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = (r_{p(j+1)}^{(\ell)} - r_{pj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} + \mathcal{O}(1/\mathtt{E}), \quad j \neq 1\\
&\mathbb{E}\left[R_{cj}^{(\ell+1)} - R_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = (r_{c(j+1)}^{(\ell)} - r_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} + \mathcal{O}(1/\mathtt{E}), \quad j \notin \{\mathtt{d}, \mathtt{d}+1\}\\
&\mathbb{E}\left[\hat{R}_{cj}^{(\ell+1)} - \hat{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = (p_j\bar{r}_{c(j+1)}^{(\ell)} + \hat{r}_{c(j+1)}^{(\ell)} - \hat{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} + \mathcal{O}(1/\mathtt{E})\\
&\mathbb{E}\left[\bar{R}_{cj}^{(\ell+1)} - \bar{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}\right] = ((1-p_j)\bar{r}_{c(j+1)}^{(\ell)} - \bar{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} + \mathcal{O}(1/\mathtt{E}), \quad j \in \{\mathtt{d}, \mathtt{d}+1\}
\end{aligned}$$

Note that $\bar{r}^{(\ell)}_{c(\mathtt{d}+2)} = r^{(\ell)}_{c(\mathtt{d}+2)}, \hat{r}^{(\ell)}_{c(\mathtt{d}+2)} = 0$. In the last two equations we are taking into account that, with probabilities $p_{\mathtt{d}+1}$ and $p_{\mathtt{d}}$ respectively, non-decodable degree-$d+2$ and degree-$d+1$ GC nodes can be turned to decodable GC nodes if they lose one edge. Again, the $\mathcal{O}(1/\mathtt{E})$ terms accounts for the fact that we compute joint probabilities over the degree of a (small) set of edges. It now also accounts for the probability of two or more removed edges connected to the same check node. We proceed similarly to compute the expected evolution of the different components when for the case Deg_{cw}, obtaining the expectations in (35)-(36). Similarly to the -1 term in (56), the indicator functions take into account the degree of the removed GC check node. From (35)-(36) we see that Condition 2) of Wormald's theorem is again satisfied. Furthermore, the expectations are actually bounded by JK. Also, it can be proved that $g_{pj}(\mathcal{G}^{(\ell)}/\mathtt{E})$, $g_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E})$, $\hat{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E})$ and $\bar{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E})$ also satisfy Condition 3), i.e., they are Lipschitz continuous on the space of possible DDs. Thus Wormald's therorem implies that with high probability we have

$$L_i^{(\ell)} = \mathtt{E} l_i(\ell/\mathtt{E}) + \mathcal{O}(\mathtt{E}^{\frac{5}{6}}), \tag{57}$$

$$R_{pj}^{(\ell)} = \mathtt{E} r_{pj}(\ell/\mathtt{E}) + \mathcal{O}(\mathtt{E}^{\frac{5}{6}}), \quad j = 1, ..., K \tag{58}$$

$$R_{cj}^{(\ell)} = \mathtt{E} r_{cj}(\ell/\mathtt{E}) + \mathcal{O}(\mathtt{E}^{\frac{5}{6}}), \quad j = 1, ..., K \text{ and } j \notin \{\mathtt{d}, \mathtt{d}+1\} \tag{59}$$

$$\hat{R}_{cj}^{(\ell)} = \mathtt{E} \hat{r}_{cj}(\ell/\mathtt{E}) + \mathcal{O}(\mathtt{E}^{\frac{5}{6}}), \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{60}$$

$$\bar{R}_{cj}^{(\ell)} = \mathtt{E} \bar{r}_{cj}(\ell/\mathtt{E}) + \mathcal{O}(\mathtt{E}^{\frac{5}{6}}), \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{61}$$

where $l_i(\ell/\mathtt{E})$, $r_{pj}(\ell/\mathtt{E})$, $r_{cj}(\ell/\mathtt{E})$, $\hat{r}_{cj}(\ell/\mathtt{E})$, $\bar{r}_{cj}(\ell/\mathtt{E})$ are the solution to

$$\frac{dl_l(\tau)}{d\tau} = f_i(\mathcal{G}^{(\ell)}/\mathtt{E}), \tag{62}$$

$$\frac{dr_{pj}(\tau)}{d\tau} = g_{pj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \tag{63}$$

$$\frac{dr_{cj}(\tau)}{d\tau} = g_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad j \notin \{\mathtt{d}, \mathtt{d}+1\} \tag{64}$$

$$\frac{d\hat{r}_{cj}(\tau)}{d\tau} = \hat{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad \frac{d\bar{r}_{cj}(\tau)}{d\tau} = \bar{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}), \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{65}$$

for the initial conditions in (24)-(27).

APPENDIX B
REFERENCE CODES GENERATOR MATRICES

Reference block codes have been found over the database [23], [24], which implements MAGMA [25] to design block codes with the largest minimum distance.

Code A: Rate-$1/2$ Hamming $(6, 3)$ linear block code

$$\mathrm{G_A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{66}$$

Code B: Rate-$1/3$ Cordaro-Wagner 2-dimensional repetition code of length 6

$$\mathrm{G_B} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{67}$$

Code C: Rate-$4/7$ Hamming (7,4) code

$$\mathrm{G_C} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{68}$$

Code D: Rate-$3/7$ linear block code

$$\mathrm{G_D} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{69}$$

Code E: Rate-$1/2$ extended $(7, 4)$-Hamming code with extra parity bit, i.e., $(8, 4)$ Hamming code. Another example is a Quasi-Cyclic $(8, 4, 4)$ code with generator matrix:

$$\mathrm{G_E} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{70}$$

Code F: Rate-$3/8$ cyclic linear block code

$$\mathrm{G_F} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{71}$$

Code G: Rate-$1/4$ Cordaro-Wagner 2-dimensional repetition code length 8

$$\mathrm{G_G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{72}$$

Code H: Rate-$11/15$ linear block code

$$\mathrm{G_H} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{73}$$

Code I: Rate-$2/3$ linear block code

$$G_I = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{74}$$

REFERENCES

[1] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533 – 547, Sept. 1981.

[2] M. Lentmaier and K. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Communications Letters*, vol. 3, no. 8, pp. 248–250, Aug 1999.

[3] G. Yue, L. Ping, and X. Wang, "Generalized Low-Density Parity-Check Codes Based on Hadamard Constraints," *IEEE Transactions on Information Theory,*, vol. 53, no. 3, pp. 1058–1079, March 2007.

[4] G. Liva, W. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Transactions on Communications*, vol. 56, no. 1, pp. 49–57, January 2008.

[5] Paolini, E. and Fossorier, M.P.C. and Chiani, M., "Generalized and Doubly Generalized LDPC Codes With Random Component Codes for the Binary Erasure Channel," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1651–1672, April 2010.

[6] D. Mitchell, M. Lentmaier, and D. Costello, "On the minimum distance of generalized spatially coupled LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT), Istanbul, Turkey*, July 2013, pp. 1874–1878.

[7] S. Abu-Surra, D. Divsalar, and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 858–886, Feb. 2011.

[8] M. Lentmaier and G. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *Proc. IEEE International Symposium on Information Theory Proceedings (ISIT), Austin, USA.*, June 2010, pp. 709–713.

[9] Y. Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012, pp. 2696–2700.

[10] M. F. Flanagan, E. Paolini, M. Chiani, and M. P. C. Fossorier, "Spectral shape of doubly-generalized ldpc codes: Efficient and exact evaluation," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7212–7228, Nov 2013.

[11] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569 –584, Feb. 2001.

[12] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5277 –5307, Dec. 2008.

[13] P. Olmos, D. Mitchell, and J. Costello, D.J., "Analyzing the finite-length performance of generalized LDPC codes," in *Proc. 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China*, June 2015, pp. 2683–2687.

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

[15] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. North-Holland Pub. Co. New York, 1977.

[16] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge, U.K., New York: Cambridge University Press, 2003.

[17] D. Mitchell, M. Lentmaier, A. Pusane, and D. Costello, "Randomly Punctured LDPC Codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 408–421, Feb 2016.

[18] Y. Wang and M. Fossorier, "Doubly generalized ldpc codes," in *2006 IEEE International Symposium on Information Theory*, July 2006, pp. 669–673.

[19] ——, "Doubly Generalized LDPC Codes over the AWGN Channel," *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1312–1319, May 2009.

[20] R. G. Gallager, *Low Density Parity Check Codes*. MIT Press, 1963.

[21] N. Miladinovic and M. Fossorier, "Generalized LDPC codes and generalized stopping sets," *IEEE Transactions on Communications*, vol. 56, no. 2, pp. 201–212, February 2008.

[22] N. C. Wormald, "Differential equations for random processes and random graphs," *Annals of Applied Probability*, vol. 5, no. 4, pp. 1217–1235, 1995.

[23] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at http://www.codetables.de, 2007, accessed on 2017-01-07.

[24] ——, "Searching for linear codes with large minimum distance," in *Discovering Mathematics with Magma — Reducing the Abstract to the Concrete*, ser. Algorithms and Computation in Mathematics, W. Bosma and J. Cannon, Eds. Heidelberg: Springer, 2006, vol. 19, pp. 287–313.

[25] W. Bosma, J. Cannon, and C. Playoust, "The Magma Algebra System I: The User Language," *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 235–265, Oct. 1997.