

Spatially Coupled Generalized LDPC Codes: Introduction and Overview

Daniel J. Costello, Jr.^{*}, David G. M. Mitchell[†], Pablo M. Olmos[‡], and Michael Lentmaier[°]

^{*}Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN, costello.2@nd.edu

[†]Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, dgmm@nmsu.edu

[‡]Signal Theory and Communications Dept., University of Carlos III in Madrid, Leganés, Spain, olmos@tsc.uc3m.es

[°]Dept. of Electrical and Information Technology, Lund University, Lund, Sweden, Michael.Lentmaier@eit.lth.se

Abstract—*Generalized low-density parity-check (GLDPC) codes are a class of LDPC codes in which the standard single parity check (SPC) constraints are replaced by more general constraints, viz., constraints defined by a linear block code. These stronger constraints typically result in improved error floor performance, due to better minimum distance and trapping set properties, at a cost of some increased decoding complexity. In this paper, we summarize some recent results on spatially coupled generalized low-density parity-check (SC-GLDPC) codes. Results are compared to GLDPC block codes and the advantages and disadvantages of SC-GLDPC codes are discussed.*

I. INTRODUCTION

*Low-density parity-check (LDPC) block codes, with iterative belief propagation (BP) decoding, were introduced by Gallager in 1963 [1] as a class of codes whose decoder implementation complexity grows only linearly with block length, in contrast to maximum likelihood (ML) and maximum a posteriori (MAP) decoding methods whose complexity typically has exponential growth. As a result of the low-density constraint on the parity-check matrix $\mathbf{H}$, the minimum distance of LDPC block codes is sub-optimal. However, Gallager showed that *regular constructions*, where the variable and check node degrees of the Tanner graph representation of $\mathbf{H}$ are fixed, maintain linear minimum distance growth with block length, i.e., they are *asymptotically good*, although their iterative decoding *thresholds* are bounded away from capacity. *Irregular constructions*, introduced by Luby et al. in 2001 [2], where the node degrees are not fixed and can be numerically optimized, have capacity-approaching thresholds, but linear distance growth is sacrificed. As a result, irregular codes perform best in the waterfall, or low signal-to-noise ratio (SNR), portion of the bit-error-rate (BER) performance curve, while regular codes perform better at high SNRs, i.e., in the *error floor* region of the BER curve.*

*Generalized LDPC (GLDPC) block codes, first proposed by Tanner in 1981 [3], are constructed by replacing some/all of the single parity-check (SPC) constraint nodes in the Tanner graph of a conventional LDPC code by more powerful generalized constraint (GC) nodes corresponding to an (n, k) linear block code. The n variable nodes connected to a GC node in the Tanner graph of a GLDPC code are then considered as the code bits of the corresponding (n, k) code, and the sub-code associated with each GC node is referred to as a *constraint code*. In message passing decoding of GLDPC codes, the constraint codes are decoded using standard block code soft-in,*

soft-out decoders which, in the case of simple constraint codes such as Hamming codes [4], can be ML or MAP decoders. GLDPC codes have several potential advantages compared to conventional SPC/LDPC codes, such as large minimum distance [4], [5] and low error floors [6].

*Spatially coupled LDPC (SC-LDPC) codes, also known as LDPC convolutional codes, were introduced by Jimenez-Felstrom and Zigangirov in 1999 [7]. SC-LDPC codes can be viewed as a sequence of LDPC block codes whose graph representations are coupled together over time, resulting in a convolutional structure with block-to-block memory. A remarkable property of SC-LDPC codes, established numerically in [8] and analytically in [9], is that their BP decoding threshold is equal to the MAP decoding threshold of the underlying LDPC block code ensemble, a phenomenon known as *threshold saturation*. In other words, the (exponential complexity) MAP decoding performance of the underlying block code can be achieved by its coupled version with (linear complexity) message passing decoding. This provides us with motivation to examine the performance of spatially coupled versions of GLDPC codes, denoted SC-GLDPC codes, in order to combine the threshold improvement of spatial coupling with the improved distance properties of generalized constraints.¹*

In this paper, we review some recent publications on SC-GLDPC codes and introduce a few new results. Protograph-based constructions and terminated SC-GLDPC codes, both of which are reviewed in Section II, are assumed throughout the paper. In Section III, we summarize the threshold analysis of terminated SC-GLDPC codes first presented in [11], followed by a review of the minimum distance analysis from [12] in Section IV. Section V begins by summarizing the approach taken to analyzing the finite-length behavior of GLDPC block codes over the binary erasure channel (BEC) with peeling decoding (PD) [13] and then presents some new results on applying the analysis to terminated SC-GLDPC codes. Concluding remarks are given in Section VI.

II. PROTOGRAPH-BASED SC-GLDPC CODES

A protograph [14] is a small bipartite graph that connects a set of n_v variable nodes $V = \{v_1, v_2, \dots, v_{n_v}\}$ to a set

¹In a recent paper [10], the authors found that, for certain *doubly-generalized* LDPC codes, in which both variable and check nodes have generalized constraints, the BP threshold is numerically indistinguishable from the MAP threshold. Hence, in these cases, no BP threshold improvement will be observed from spatial coupling.

of n_c constraint nodes $C = \{c_1, c_2, \dots, c_{n_c}\}$ by a set of edges E . In a protograph-based GLDPC code ensemble, each constraint node c_i can represent an arbitrary block constraint code of length n^{c_i} with m^{c_i} linearly independent parity-check equations. The *design rate* of the GLDPC code ensemble is then given by

$$R = 1 - \frac{\sum_{i=1}^{n_c} m^{c_i}}{n_v}. \quad (1)$$

A protograph can be represented by means of an $n_c \times n_v$ bi-adjacency matrix $\mathbf{B}$, which is called the *base matrix* of the protograph. The nonnegative integer entry B_{ij} in row i and column j of $\mathbf{B}$ is equal to the number of edges that connect nodes c_i and v_j . In order to construct ensembles of protograph-based GLDPC codes, a protograph can be interpreted as a template for the Tanner graph of a derived code, which can be obtained by a *copy-and-permute* or *graph lifting* operation [14]. In matrix form, the protograph is lifted by replacing each nonzero entry B_{ij} of $\mathbf{B}$ with a summation of B_{ij} non-overlapping permutation matrices of size $M \times M$, thereby creating an $Mn_c \times Mn_v$ parity-check matrix $\mathbf{H}$ of a GLDPC code. Each row in the i^{th} set of M rows of $\mathbf{H}$ must satisfy the constraint associated with constraint node c_i , where the length n^{c_i} of the i^{th} constraint code equals the number of nonzero entries in the i^{th} row of $\mathbf{B}$, and the constraint applies to the positions in a row with nonzero entries.² Allowing the permutations to vary over all $M!$ possible choices results in an ensemble of GLDPC block codes.

Example 1: Fig. 1 displays the protograph of a $(2, 7)$ -regular GLDPC block code with base matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (2)$$

If we suppose both the constraint nodes are $(7, 4)$ Hamming codes with parity-check matrix

$$\mathbf{H}_{c_1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad (3)$$

where the constraint code length is $n^{c_1} = 7$ and the row rank is $m^{c_1} = 3$, then the resulting ensemble has design rate $R = 1/7$. Note that even though both constraints are defined by the same $(7, 4)$ Hamming code, a different ordering of columns can be used. In Fig. 1, the column of $\mathbf{H}_{c_1}$ that the variable node is connected to is shown on the edge. $\square$

A. Convolutional Protographs

An unterminated SC-GLDPC code can be described by a *convolutional protograph* [15] with base matrix

$$\mathbf{B}_{[0, \infty]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \mathbf{B}_1 & \ddots & \\ \mathbf{B}_w & \vdots & \ddots & \\ & \mathbf{B}_w & & \ddots \end{bmatrix}, \quad (4)$$

²Strictly speaking, $\mathbf{H}$ is not a true parity-check matrix, since each row in the i^{th} set of M rows of $\mathbf{H}$ corresponds to m^{c_i} parity checks.

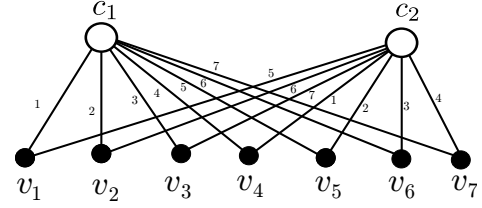


Fig. 1: Protograph of a $(2, 7)$ -regular GLDPC block code. The white circles represent generalized constraint nodes and the black circles represent variable nodes.

where w denotes the *syndrome former memory* or *coupling width* of the code and the $b_c \times b_v$ *component base matrices* $\mathbf{B}_i$, $i = 0, 1, \dots, w$, represent the edge connections from the b_v variable nodes at time t to the b_c (generalized) constraint nodes at time $t + i$. An ensemble of (in general) time-varying SC-GLDPC codes can then be formed from $\mathbf{B}_{[0, \infty]}$ using the protograph construction method described above with lifting factor M . The *decoding constraint length* of the resulting ensemble is given by $\nu_s = (w + 1)Mb_v$, and at each time instant t the encoder creates a block $\mathbf{v}_t$ of Mb_v symbols resulting in the unterminated code sequence $\mathbf{v} = [\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_t, \dots]$.

Starting from the base matrix $\mathbf{B}$ of a block code ensemble, one can construct SC-GLDPC code ensembles with the same variable and check node degrees as $\mathbf{B}$. This is achieved by an *edge spreading* procedure (see [15] for details) that divides the edges connected to each variable node in the base matrix $\mathbf{B}$ among $w + 1$ component base matrices $\mathbf{B}_i$, $i = 0, 1, \dots, w$, such that the condition $\mathbf{B}_0 + \mathbf{B}_1 + \dots + \mathbf{B}_w = \mathbf{B}$ is satisfied.

Example 2: For $w = 1$, we can apply the edge spreading technique to the $(2, 7)$ -regular block base matrix in (2) to obtain the following component base matrices

$$\mathbf{B}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5)$$

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (6)$$

The convolutional protograph associated with the resulting base matrix $\mathbf{B}_{[0, \infty]}$ defined in (4) is shown in Fig. 2. We choose the upper and lower constraint nodes at each time instant to correspond to the $(7, 4)$ Hamming codes with parity check matrix $\mathbf{H}_{c_1}$ from (3). Note that the column ordering is different for c_1 and c_2 as indicated on the edges. $\square$

B. Terminated SC-GLDPC Codes

Suppose that we start the convolutional code with parity-check matrix defined in (4) at time $t = 0$ and terminate it after L time instants (corresponding to $t = L$ in Fig. 2). The resulting finite-length base matrix is then given by

$$\mathbf{B}_{[0, L-1]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \mathbf{B}_1 & \ddots & \\ \mathbf{B}_w & \vdots & \ddots & \mathbf{B}_0 \\ & \mathbf{B}_w & & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_w \end{bmatrix}_{(L+w)b_c \times Lb_v}. \quad (7)$$

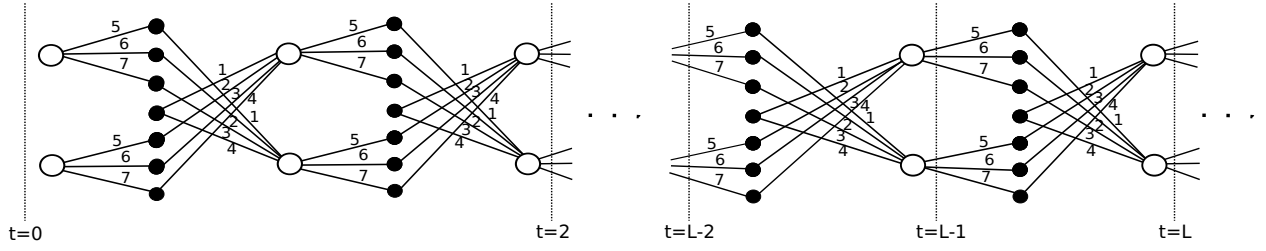


Fig. 2: Protograph of a $(2, 7)$ -regular SC-GLDPC code ensemble.

The matrix $\mathbf{B}_{[0,L-1]}$ is then the base matrix of a *terminated SC-GLDPC code*. This terminated protograph is slightly irregular, with lower constraint node degrees at each end.

The reduced degree constraint nodes at each end of the convolutional protograph are associated with shortened constraint codes, in which the symbols corresponding to the missing edges are removed. Such a code shortening is equivalent to fixing these removed symbols and assigning an infinite reliability to them. Note that the variable node degrees are not affected by termination.

The parity-check matrix $\mathbf{H}_{[0,L-1]}$ of the terminated SC-GLDPC code derived from $\mathbf{B}_{[0,L-1]}$ by lifting with some factor M has $Mb_v L$ columns and $(L+w)Mb_c$ rows. It follows that the rate of the terminated SC-GLDPC code is equal to

$$R_L = 1 - \frac{(L+w)b_c m^c - \Delta}{Lb_v}, \quad (8)$$

where m^c denotes the (constant) number of parity-checks of each constraint code and $\Delta \geq 0$ accounts for a slight rate increase due to the shortened constraint codes.³ If $\mathbf{H}_{[0,L-1]}$ has full rank, the rate increase parameter is $\Delta = 0$. However, the shortened constraint codes at the ends of the graph can cause a reduced rank for $\mathbf{H}_{[0,L-1]}$, which slightly increases R_L . In this case, $\Delta > 0$ and depends on both the particular constraint code chosen and the assignment of edges to the columns of $\mathbf{H}_{c_i}$. As $L \rightarrow \infty$, the rate R_L converges to the design rate $R = 1 - b_c m^c / b_v$ of the underlying GLDPC block code with base matrix $\mathbf{B}$.⁴

III. THRESHOLD ANALYSIS OF TERMINATED SC-GLDPC CODES

Assume that we start encoding at time $t = 0$ and terminate after L time instants. As a result we obtain the block protograph $\mathbf{B}_{[0,L-1]}$ from (7). These terminated SC-GLDPC codes can be interpreted as GLDPC block codes that inherit the structure of the convolutional codes. The length of these codes depends not only on the lifting factor M but also on the termination factor L . For a fixed L , the BEC density evolution thresholds $\varepsilon_{L,\text{BP}}$ corresponding to codes with base matrix $\mathbf{B}_{[0,L-1]}$ can be estimated using the methods of [11] for different channel parameters ε . The resulting thresholds

³We assume here that each generalized constraint node c_i in the block protograph has m^c parity-checks.

⁴We note here that the $(L+w)Mb_c$ rows of $\mathbf{H}_{[0,L-1]}$ should be viewed as $(L+w)b_c$ groups of rows, with M entries in each group, that are decoded according to the same constraint code with m^c rows.

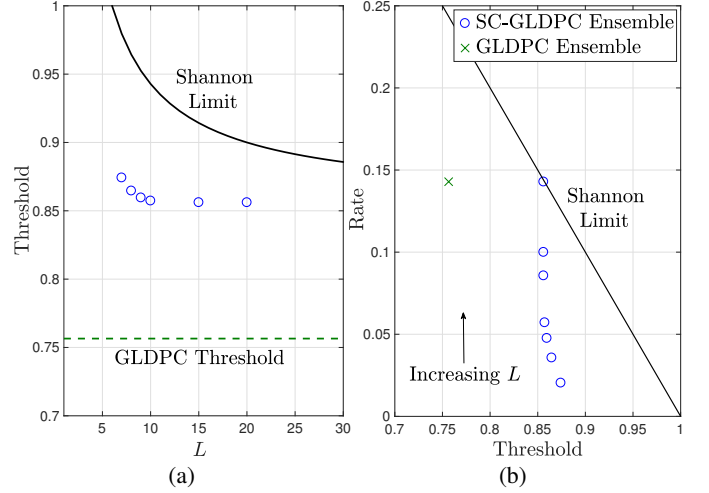


Fig. 3: (a) BP decoding thresholds as functions of the termination factor L and (b) BP decoding thresholds versus code rate R_L .

for the $(2, 7)$ -regular $w = 1$ ensemble with $(7, 4)$ Hamming constraint codes from Example 2 are shown in Fig. 3(a) for different termination factors L , and the thresholds versus code rate R_L are shown in Fig. 3(b). Analogously to SC-LDPC codes (see [15]) with SPC constraints, we observe that as $L \rightarrow \infty$ the BP threshold numerically coincides with the upper bound on the ML decoding threshold of the corresponding block code ensemble, thus exhibiting the *threshold saturation* phenomenon (see [8], [9]). Large L can be realistic in conjunction with window based decoders, like those suggested in [16], where decoding delay and storage requirements depend on the window size W , which is independent of the length L (typically $W \ll L$) of the transmitted code sequences. For shorter L , which introduces rate loss, BP decoding of terminated SC-GLDPC codes is suboptimal but still provides a flexible adjustment between code rate and threshold (see Fig. 3(b)).

IV. MINIMUM DISTANCE ANALYSIS OF TERMINATED SC-GLDPC CODES

From the convolutional protograph with base matrix $\mathbf{B}_{[0,\infty]}$ in (4), we can form a periodically time-varying M -fold graph cover with period T by choosing, for the $b_c \times b_v$ submatrices $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_w$ in the first T columns of $\mathbf{B}_{[0,\infty]}$, a set of $M \times M$ permutation matrices randomly and independently to form $Mb_c \times Mb_v$ submatrices $\mathbf{H}_0(t), \mathbf{H}_1(t+1), \dots, \mathbf{H}_w(t+w)$, respectively, for $t = 0, 1, \dots, T-1$. These submatrices can then

be repeated periodically (indefinitely) to form a convolutional parity-check matrix $\mathbf{H}_{[0,\infty]}$ such that $\mathbf{H}_i(t+T) = \mathbf{H}_i(t)$, $\forall i, t$. An ensemble of periodically time-varying SC-GLDPC codes with period T , design rate $R = 1 - M^c b_c / M b_v = 1 - m^c b_c / b_v$, and decoding constraint length $\nu_s = M(w+1)b_v$ can then be derived by letting the permutation matrices used to form $\mathbf{H}_0(t), \mathbf{H}_1(t+1), \dots, \mathbf{H}_w(t+w)$, for $t = 0, 1, \dots, T-1$, vary over the $M!$ choices of an $M \times M$ permutation matrix.

In [17], Abu-Surra, Divsalar, and Ryan presented a technique to calculate the average weight enumerator and asymptotic spectral shape function for protograph-based GLDPC block code ensembles. The spectral shape function can be used to test if an ensemble is *asymptotically good*, i.e., if the minimum distance typical of most members of the ensemble is at least as large as $\delta_{\min} n$, where $\delta_{\min}$ is the *minimum distance growth rate* of the ensemble and n is the block length.

Example 3: Consider the $(2, 7)$ -regular GLDPC block code protograph with the all-ones base matrix $\mathbf{B}$ from (2) and the two generalized constraint nodes shown in Fig. 1. If the constraint codes are $(7, 4)$ Hamming codes with parity-check matrix $\mathbf{H}_{c_1}$ from (3), then the resulting ensemble has design rate $R = 1/7$, is asymptotically good, and has growth rate $\delta_{\min} = 0.186$ [17]. $\square$

We now consider the associated $(2, 7)$ -regular terminated SC-GLDPC code ensemble from Example 2, whose protograph is shown in Fig. 2. Since terminated SC-GLDPC codes can be viewed as block codes of length $M b_v L$ and rate given by (8), the methods of [17], properly modified, can be used to evaluate their asymptotic weight enumerators (see [12] for details). Fig. 4 shows the asymptotic spectral shape functions for the SC-GLDPC code ensembles with termination factors $L = 7, 8, 10, 12, 14, 16, 18$, and 20 . Also shown are the asymptotic spectral shape functions for “random” codes with the corresponding rates R_L calculated using (see [1])

$$r(\delta) = H(\delta) - (1 - R_L) \ln(2), \quad (9)$$

where $H(\delta) = -(1 - \delta) \ln(1 - \delta) - \delta \ln(\delta)$. We observe that the SC-GLDPC code ensembles are asymptotically good and have relatively large minimum distance growth rates, ranging from about 25% to 65% of the random coding growth rates. This indicates that long codes chosen from these ensembles have, with probability near one, a large minimum distance. As L increases, R_L approaches the design rate $R = 1/7$ of the underlying GLDPC block code and the minimum distance growth rate decreases, as was also observed in the case of SC-LDPC codes with SPC constraints (see [15]).

While large minimum distance growth rates are indicative of good ML decoding performance, when predicting the iterative decoding performance of a code ensemble in the high SNR region other graphical objects such as pseudocodewords, trapping sets, absorbing sets, etc., come into effect. Based on results from the SPC case [18], we expect SC-GLDPC codes with large minimum distance growth rates to also have large trapping set growth rates, indicating good iterative decoding performance in the high SNR region.

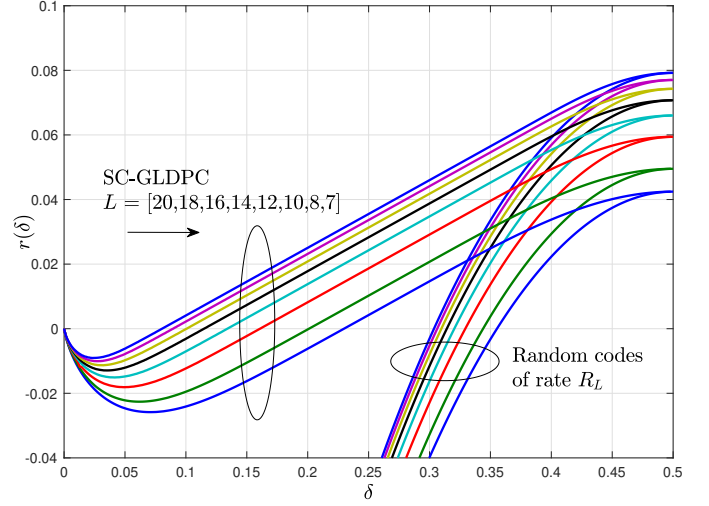


Fig. 4: Spectral shape functions of SC-GLDPC code ensembles and random linear codes of the corresponding rate.

V. FINITE-LENGTH ANALYSIS OF TERMINATED SC-GLDPC CODES

Peeling decoding (PD) is a simple algorithm for LDPC codes over the BEC that iteratively removes a degree-one check node in the graph along with the variable node attached to it and all the edges connected to these two nodes. We now briefly describe an extension of PD to GLDPC codes, referred to as generalized peeling decoding (GPD) (see [13]).

Initially, variable nodes of $\mathbf{H}$ and their attached edges are removed from the graph with probability $(1 - \epsilon)$. After initialization, the residual graph contains constraint nodes with types that are not included in $\mathcal{F}_c$, the set of constraint node types in the original graph, but the set of variable node types $\mathcal{F}_v$ remains the same. Given a constraint node of type $\mathbf{c} \in \mathcal{F}_c$, define $\mathcal{D}(\mathbf{c})$ as the set of all residual constraint node types that might appear in the graph when edges are removed from a constraint node of type $\mathbf{c}$ ($\mathbf{c} \in \mathcal{D}(\mathbf{c})$). The extended set of all possible constraint node types which are present in the residual graph after GPD initialization is then given by $\overline{\mathcal{F}}_c \doteq \bigcup_{\mathbf{c} \in \mathcal{F}_c} \mathcal{D}(\mathbf{c})$.

Example 4: In the $(2, 7)$ -regular protograph of Fig. 1, we have two constraint node types: $\mathbf{c}_1$, denoting the set of edges connecting constraint node 1 to each of the 7 variable nodes, and $\mathbf{c}_2$, denoting the set of edges connecting constraint node 2 to each of the 7 variable nodes. Corresponding to each of these types, $2^7 = 128$ residual types can appear in the graph when edges are removed. Thus, in total, $\overline{\mathcal{F}}_c$ contains 256 residual types. $\square$

According to the above definitions, the expected degree distribution (DD) after initialization can be expressed as follows:

$$L_{\mathbf{d}}(0) = \epsilon L_{\mathbf{d}}, \quad R_{\mathbf{c}}(0) = \sum_{\substack{\mathbf{q} \in \mathcal{F}_c \\ \mathbf{c}' \in \mathcal{D}(\mathbf{q})}} R_{\mathbf{q}} \epsilon^{|\mathbf{c}'|} (1 - \epsilon)^{|\mathbf{q}| - |\mathbf{c}'|},$$

for $\mathbf{d} \in \mathcal{F}_v$ and $\mathbf{c} \in \overline{\mathcal{F}}_c$, where $L_{\mathbf{d}}$ (respectively $R_{\mathbf{c}}$) is the number of variable (constraint) nodes of type $\mathbf{d}$ ($\mathbf{c}$) in the original graph, $L_{\mathbf{d}}(0)$ ($R_{\mathbf{c}}(0)$) is this number after GPD initialization, and $|\mathbf{c}'|$ is the number of edges in $\mathbf{c}'$.

We now define the normalized DD at time τ as follows:

$$\tau \doteq \frac{\ell}{M}, \quad r_c(\tau) \doteq \frac{R_c(\tau)}{M}, \quad l_d(\tau) \doteq \frac{L_d(\tau)}{M}, \quad (10)$$

where ℓ is the GPD iteration index, $R_c(\tau)$ ($L_d(\tau)$) is the number of constraint (variable) nodes in the graph of type $c(d)$ at time τ , and M is the lifting factor. Following the methodology developed in [19] to analyze the BEC finite-length performance of LDPC block codes, we can investigate the BEC finite-length performance of GLDPC codes by analyzing the statistical evolution of the DD in (10) during the decoding process. As shown in [19], the GPD threshold is defined as the maximum value of ϵ for which the expected fraction of decodable constraint nodes

$$\hat{a}(\tau) \doteq \sum_{c \in \mathcal{A}} \hat{r}_c(\tau), \quad (11)$$

is positive for any $\tau \in [0, \epsilon)$, where $\hat{r}_c(\tau)$ is the expected value of $r_c(\tau)$, $\hat{a}(\tau)$ is the mean of the random process

$$a(\tau) \doteq \sum_{c \in \mathcal{A}} r_c(\tau), \quad (12)$$

and $\mathcal{A}$ is the set of all decodable constraint node types, which depends on the erasure correcting capability of the constraint codes. Similarly, we can compute the expected fraction of variable nodes in the graph at any time τ , denoted by $\hat{v}(\tau)$, as follows:

$$\hat{v}(\tau) \doteq \sum_{d \in \mathcal{F}_v} \hat{l}_d(\tau), \quad (13)$$

where $\hat{l}_d(\tau)$ is the expected value of $l_d(\tau)$. $\hat{r}_c(\tau)$ and $\hat{l}_d(\tau)$ can be computed as the solution to a system of differential equations (see [19] for details) and then used to determine the quantities needed to assess the finite-length performance of the code. We refer to *critical points* as the points in time for which $\hat{a}(\tau)$ has a local minima. As shown in [19], the average (over the ensemble of codes) error probability is dominated by the probability that the process $a(\tau)$ *survives*, i.e., does not go to zero around the critical points. Therefore, characterizing the critical points and the expected fraction of decodable constraint nodes in the graph at those points in time are the parameters needed to determine the GLDPC code finite-length performance.

With the tools described above, we can now investigate the asymptotic and finite-length performance of GLDPC block code and terminated SC-GLDPC code ensembles.

Example 5: Consider the (2, 7)-regular GLDPC block code ensemble of Fig. 1 with ML-decoded Hamming (7, 4) constraint codes. The design rate of this ensemble is $R = 1/7$. All constraint node types with one or two erasures can be decoded, as well as some constraint node types with three erasures. Fig. 5 shows the evolution of the expected fraction of decodable constraint nodes $\hat{a}(\tau)$ versus the expected fraction of variable nodes $\hat{v}(\tau)$ in the graph for different ϵ values.⁵

⁵Note that the time variable τ runs backwards in this figure (right to left), in the sense that small values of τ correspond to $\hat{v}(\tau)$ on the right, where the graph still contains a relatively large fraction of variable nodes, whereas large values of τ correspond to small values of $\hat{v}(\tau)$ on the left.

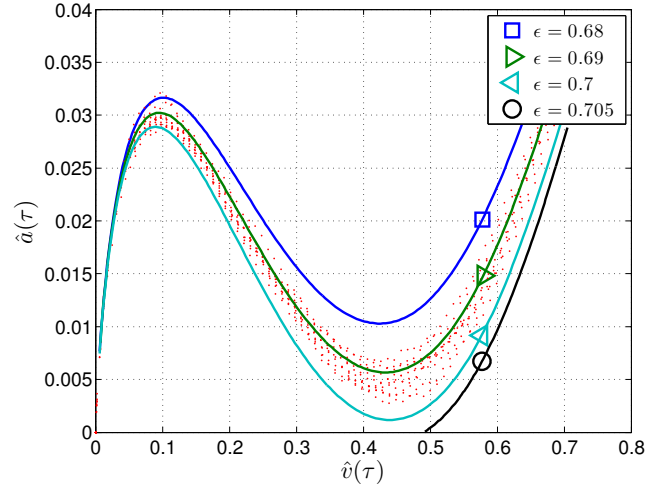


Fig. 5: Evolution of the expected fraction of decodable constraint nodes $\hat{a}(\tau)$ in the residual graph during iterations of the GPD for the (2, 7)-regular GLDPC block code ensemble with (7, 4) Hamming constraint codes decoded using an ML decoder. The dotted curves represent simulated trajectories for $a(\tau)$ computed for $\epsilon = 0.69$ with lifting factor $M = 4000$.

We also include a set of 10 simulated trajectories for $a(\tau)$ for $\epsilon = 0.69$ to demonstrate that they concentrate around the predicted mean. Note first that $\hat{a}(\tau)$ has a single critical point at $\hat{v}(\tau^*) \approx 0.43$. Indeed, we can compute the threshold ϵ^* as the maximum ϵ value for which the minimum is exactly zero. In this case we obtain $\epsilon^* \approx 0.7025$. $\square$

The finite-length error probability is dominated by the statistics of $a(\tau)$ around τ^* . Following [19], for each n and ϵ pair, we can estimate the finite-length error rate as

$$P_{\text{Block}} \sim \mathcal{Q} \left(\frac{\hat{a}(\tau^*)}{\sqrt{\text{Var}[a(\tau^*)]}} \right), \quad (14)$$

where $\hat{a}(\tau^*)$ is the expected value of $a(\tau)$ at τ^* , $\text{Var}[a(\tau^*)]$ represents its variance, and we use Monte Carlo simulation to estimate $\text{Var}[a(\tau^*)]$ for a particular (n, ϵ) pair. In [19], the authors showed that the ratio of the expected number of degree-one constraint nodes to the standard deviation at the critical point approximately scales as $\alpha\sqrt{n}(\epsilon^* - \epsilon)$, where α is a *scaling parameter* that only depends on the DD. In the GLDPC case, the simulated trajectories for $a(\tau)$ suggest that the same scaling holds and that the performance for any pair (n, ϵ) can be estimated as $P_{\text{Block}} \sim \mathcal{Q}(\alpha\sqrt{n}(\epsilon^* - \epsilon))$.

Following a similar procedure, we can analyze the finite-length behavior of terminated SC-GLDPC codes. Once the SC-GLDPC base matrix $\mathbf{B}_{[0, L-1]}$ is defined, we can use the same analysis described above, including the computation of the expectations $\hat{a}(\tau)$ and $\hat{v}(\tau)$ in (11) and (13). In Figure 6, we show the evolution of the expected fraction $\hat{a}(\tau)$ of decodable check nodes during iterations of the GPD for the (2, 7)-regular terminated SC-GLDPC code ensemble (corresponding to the GLDPC block code ensemble of Example 5) with $L = 20, 30$, and 50 for a channel parameter $\epsilon = 0.75$. Unlike the GLDPC block code, the expected evolution $\hat{a}(\tau)$ displays a constant evolution or *critical phase* of decoding that corresponds to a decoding wave traveling towards the central positions of the graph. Further, the critical value $\hat{a}(\tau^*)$ during such a phase

does not depend on L , and the length of the critical phase is roughly proportional to L . Similar effects were first described in [20] for non-generalized terminated SC-LDPC codes. Based on this evidence, it is expected that the survival probability of the $a(\tau)$ process during the critical phase follows a scaling law of the same form as the one proposed in [20], and thus the block error probability P_{Block} can be estimated as follows

$$P_{\text{Block}} \approx 1 - \exp \left(- \frac{\nu L}{\int_0^{\alpha \sqrt{M}(\epsilon^* - \epsilon)} \Phi(z) e^{\frac{1}{2} z^2} dz} \right), \quad (15)$$

where $\Phi(z)$ is the c.d.f. of the standard Gaussian distribution, $\mathcal{N}(0,1)$, νL is the length of the critical phase, and $\alpha \sqrt{M}(\epsilon^* - \epsilon)$ corresponds to the ratio of the expected number of decodable constraint nodes during the critical phase to the standard deviation of $a(\tau)$. Both ν and α are parameters that depend on the underlying GLDPC block code and the edge spreading.

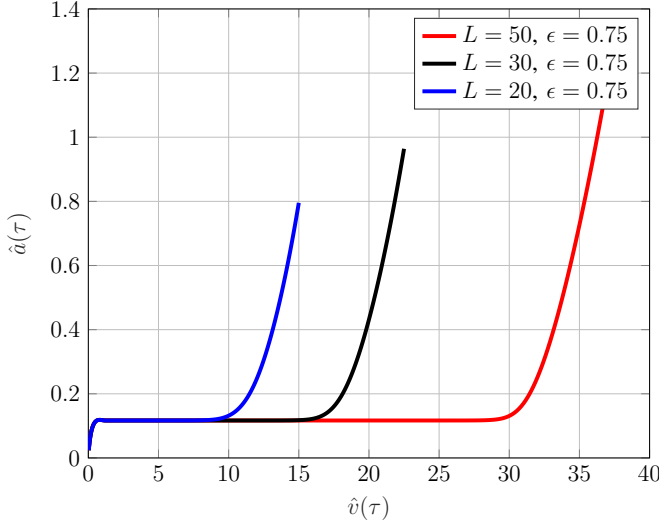


Fig. 6: Evolution of the expected fraction of decodable constraint nodes $\hat{a}(\tau)$ in the residual graph during iterations of the GPD for the $(2,7)$ -regular SC-GLDPC code ensemble with $L = 20, 30$, and 50 and $(7,4)$ Hamming constraint codes decoded using an ML decoder.

VI. CONCLUDING REMARKS

In this paper we summarized both asymptotic and finite-length results for SC-GLDPC codes. Specifically, terminated SC-GLDPC code ensembles were shown to achieve threshold saturation, thus assuring them of having better waterfall performance than their underlying GLDPC block codes. They were then shown to be asymptotically good and to possess large minimum distance growth rates, thus assuring them of also having excellent error floor performance. Finally, terminated SC-GLDPC codes were shown to outperform their GLDPC block code counterparts in the finite length regime. These improvements are achieved at the expense of a modest increase in decoding complexity, depending on the particular constraint

codes and decoders chosen, albeit with the advantage of a typically smaller number of message passing iterations.

ACKNOWLEDGMENT

This material is based on work supported in part by the National Science Foundation under Grant No. ECCS-1710920 and in part by the Spanish Ministry of Science, Innovation and University under grant TEC2016-78434-C3-3-R (AEI/FEDER, EU).

REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1963.
- [2] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.
- [3] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. on Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.
- [4] M. Lentmaier and K. Sh. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Comm. Letters*, vol. 8, no. 8, pp. 248–250, Aug. 1999.
- [5] J. J. Boutros, O. Pothier, and G. Zémor, "Generalized low density Tanner codes," in *Proc. IEEE Int. Conf. Comm.*, Vancouver, Canada, June 1999.
- [6] G. Liva, W. E. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Trans. on Comm.*, vol. 56, no. 1, pp. 49–57, Jan. 2008.
- [7] A. Jiménez Felström and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrices," *IEEE Trans. on Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sept. 1999.
- [8] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. on Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.
- [9] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. on Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [10] A. D. Yardi, I. Andriyanova, and C. Poulliat, "EBP-GEXIT charts over the binary-input AWGN channel for generalized and doubly-generalized LDPC codes," in *Proc. IEEE Int. Symp. on Inf. Theory*, Jun. 2018, pp. 496–500.
- [11] M. Lentmaier and G. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *Proc. IEEE Int. Symp. on Inf. Theory*, Austin, TX, July 2010.
- [12] D. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "On the minimum distance of generalized spatially coupled LDPC codes," in *Proc. IEEE Int. Symp. on Inf. Theory*, Istanbul, Turkey, 2013, pp. 1874–1878.
- [13] P. Olmos, D. Mitchell, and D. Costello, "Analyzing the finite-length performance of generalized LDPC codes," in *Proc. IEEE Int. Symp. on Inf. Theory*, Hong Kong, China, July 2015, pp. 2683–2687.
- [14] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," Jet Propulsion Laboratory, Pasadena, CA, INP Progress Report 42-154, Aug. 2003.
- [15] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. on Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.
- [16] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. on Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [17] S. Abu-Surra, D. Divsalar, and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Trans. on Inf. Theory*, vol. 57, no. 2, pp. 858–886, Feb. 2011.
- [18] D. G. M. Mitchell, A. E. Pusane, and D. J. Costello, Jr., "Minimum distance and trapping set analysis of protograph-based LDPC convolutional codes," *IEEE Trans. on Inf. Theory*, vol. 59, no. 1, pp. 254–281, Jan. 2013.
- [19] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Trans. on Inf. Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.
- [20] P. Olmos and R. Urbanke, "A scaling law to predict the finite-length performance of spatially-coupled LDPC codes," *IEEE Trans. on Inf. Theory*, vol. 61, no. 6, pp. 3164–3184, June 2015.