Density Evolution Analysis of Protograph-Based Braided Block Codes on the Erasure Channel

Lentmaier, Michael; Nöthen, Benedikt; Fettweis, Gerhard

Published in:
2010 International ITG Conference on Source and Channel Coding (SCC)

2010

Density Evolution Analysis of Protograph-Based Braided Block Codes on the Erasure Channel

Michael Lentmaier, Benedikt Nöthen, and Gerhard P. Fettweis
Vodafone Chair Mobile Communications Systems
Dresden University of Technology (TU Dresden), 01062 Dresden, Germany
Emails: {michael.lentmaier, benedikt.noethen, fettweis}@ifn.et.tu-dresden.de

***Abstract*—A threshold analysis of terminated braided block codes (BBCs) is presented for the binary erasure channel. Protograph based BBCs (PG-BBCs) are considered, which can be derived by lifting the Tanner graph of a tightly braided block code. They can be interpreted as GLDPC convolutional codes that are structured in a particular way. It is shown that the terminated PG-BBCs have better thresholds than their GLDPC block code counterparts. Simulation results are presented to confirm the computed asymptotic thresholds.**

I. Introduction

Braided block codes (BBCs) are a convolutional version of Elias' product codes [1], where each code symbol is protected by two component block codes [2] [3]. Similar to Gallager's LDPC codes [4], sparsity can be introduced into the structure of BBCs to construct codes of arbitrary length and memory without changing the component codes. Such sparse BBCs can be interpreted as LDPC convolutional codes (LDPC CCs) [5] with stronger component codes, i.e., *generalized* LDPC (GLDPC) convolutional codes.

Like GLDPC block codes [6], BBCs can be decoded iteratively with the belief propagation (BP) algorithm. In this case, the individual component codes are decoded by *a posteriori probability* (APP) decoders and the resulting reliability information about the symbols is exchanged among them during the decoding iterations. Due to their convolutional structure, BBCs are suitable for continuous transmission by means of a pipeline decoding structure, analogous to LDPC CCs [5]. Simulation results show that BBCs with relatively small memory come remarkably close to the asymptotic GLDPC threshold (see Fig. 1). On the other hand, it can be observed that *terminated* BBCs can even outperform the thresholds of the corresponding regular GLDPC codes [3].

The aim of this paper is to show that terminated BBCs indeed have better thresholds than their GLDPC counterparts. Such a threshold analysis becomes feasible for the binary erasure channel (BEC), where explicit input/output transfer functions for the component APP decoders are available [7]. Using these functions we are able to generalize the analysis of terminated LDPC CCs in [8] [9] to the BBCs. We start with an introduction to BBCs in Section II. Then, in Section III, (G)LDPC code ensembles based on protographs are introduced and it is demonstrated how exact density evolution equations can be used for the computation of their BEC thresholds. Using these equations, a threshold analysis is performed in Section IV for terminated, protograph based BBCs. Section V concludes the paper.

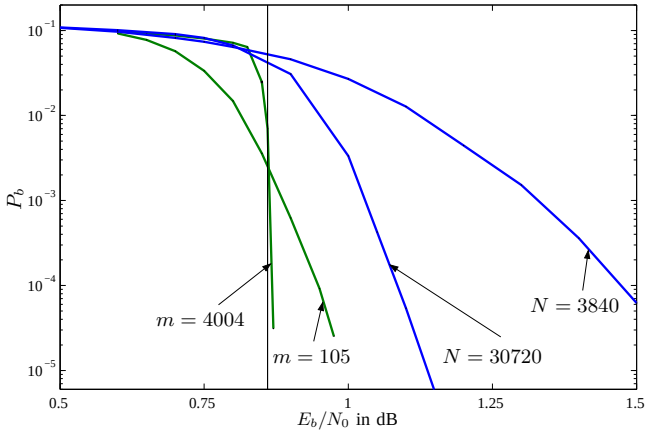

Fig. 1. Simulation results for continuous BBCs of memory m, transmitted over an additive white Gaussian noise (AWGN) channel. Length N GLDPC block codes and their threshold (vertical line) are shown for comparison. All codes are based on (15,11) Hamming component codes, resulting in the code rate $R = 7/15$.

II. Braided Block Codes

BBCs can be defined by means of an infinite, diagonally shaped array that contains the different code symbols. Analogously to product codes [1], each code symbol is protected by one horizontal and one vertical component code. The array of a tightly braided block code (TBBC), where the width is equal to the length of the component codes, is illustrated in Fig. 2. The gray-shaded area contains the already encoded symbols. At time instant t the horizontal component encoder operates in row t of the array. Using the symbols $\tilde{\mathbf{v}}_t^{(1)}$ from the left of the main diagonal together with the new information symbols $[u_{t,0}, \mathbf{u}_t^{(1)}]$, it produces the parity-check symbols $\hat{\mathbf{v}}_t^{(1)}$ and stores them in the right part of row t. Analogously, the vertical encoder produces the parity-check symbols $\hat{\mathbf{v}}_t^{(2)}$ in column t by encoding the symbols $\tilde{\mathbf{v}}_t^{(2)}$ together with the new information symbols $[u_{t,0}, \mathbf{u}_t^{(2)}]$ and stores them in the lower part of column t. Repeating this procedure, the encoder slides along the main diagonal of the array as new information symbols arrive.

The output of the TBBC encoder is the infinite code

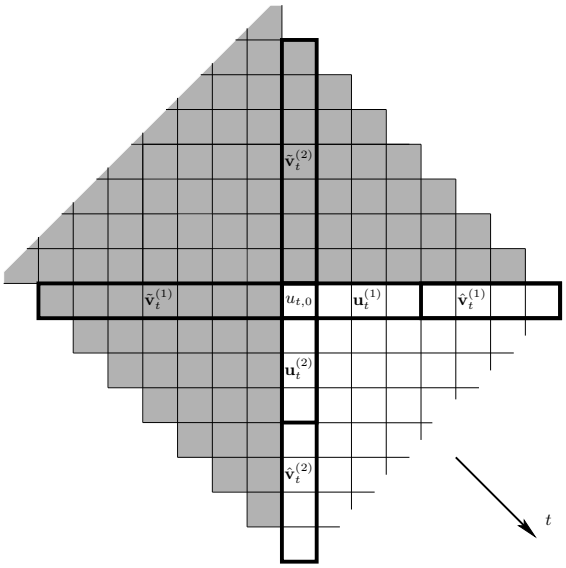

Fig. 2. Array representation of a TBBC with (15,11) Hamming component codes.

sequence $\mathbf{v} = [\ldots, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t, \ldots]$, where

$$\mathbf{v}_t = [u_{t,0}, \mathbf{u}_t^{(1)}, \mathbf{u}_t^{(2)}, \hat{\mathbf{v}}_t^{(1)}, \hat{\mathbf{v}}_t^{(2)}] \tag{1}$$

are the symbols produced at time t. The rate of a TBBC is equal to

$$R = R^{(1)} + R^{(2)} - 1 \ , \tag{2}$$

where $R^{(1)}$ and $R^{(2)}$ denote the rates of the horizontal and vertical component codes, respectively. Since TBBCs are convolutional codes, the code symbols $\mathbf{v}_t$ depend not only on the new information symbols arriving at time t but also on the code symbols $\mathbf{v}_{t-1}, \ldots, \mathbf{v}_{t-m_{\mathrm{cc}}}$ of previous time instants. The value m_{cc} defines the *memory* of the TBBC.

Example 1: Assume that both the horizontal and the vertical component codes are (15,11) Hamming codes. Then the input vectors $[\tilde{\mathbf{v}}_t^{(i)}, u_{t,0}, \mathbf{u}_t^{(i)}]$, $i = 1, 2$, of the encoders have length 11 and the vectors $\hat{\mathbf{v}}_t^{(i)}$ consist of 4 parity symbols. Since $\tilde{\mathbf{v}}_t^{(i)}$ and $[\mathbf{u}_t^{(i)}, \hat{\mathbf{v}}_t^{(i)}]$ must have the same length, it follows that the vectors $\mathbf{u}_t^{(i)}$ have length 3. The TBBC has rate $R = 7/15$ and memory $m_{\mathrm{cc}} = 7$. The array depicted in Fig. 2 corresponds to this case. □

Like a (G)LDPC code, a TBBC can be described by a Tanner graph [10]. Then each row and each column of the array is represented by a constraint node and each symbol by a variable node. A variable node is connected to a constraint node if the associated symbol participates in the corresponding component code. The Tanner graph of a TBBC with (7,4) Hamming component codes is illustrated in Fig. 3. Its girth is equal to eight, which follows from the structure of the array and is true for any TBBC.

In a sparsely braided block code (SBBC), the symbols in the array do not have to be adjacent but can be spread apart from each other. Allowing such a sparse structure, the width of the array and hence the memory of the SBBC can be increased independently of the lengths of the component codes. In this paper, we take a different approach as in [3] and use the Tanner graph of a TBBC as protograph for constructing codes of arbitrary sparsity.

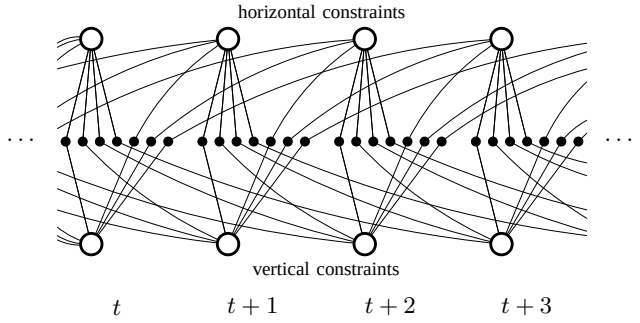

Fig. 3. Tanner graph of a TBBC with (7,4) Hamming component codes. The nodes are grouped according to the time instant at which the code symbols are generated.

III. Density Evolution for Protograph Ensembles

A. LDPC Codes

Structured LDPC codes with parity-check matrices that are composed of individual permutation matrices already appeared in Gallager's original work [4]. For such code ensembles he proposed an algorithm to construct LDPC codes of arbitrarily large girth, provided that the block length is chosen sufficiently large. In [11] it was observed that the asymptotic minimum distance ratios of these structured ensembles are the same as for the ensembles that cannot be divided into permutation matrices. Quasi-cyclic LDPC codes can be obtained if the permutation matrices are restricted to be circulants [12]. In order to obtain structured irregular LDPC codes, some of the permutation matrices can be replaced by all-zero matrices [13]. The structure of a permutation based LDPC ensemble can be represented in a compact form by means of a protograph [14].

A protograph is a bipartite graph consisting of a set of variable nodes V_n with degree J_n, $n = 1, \ldots, N_P$, a set of constraint nodes C_m with degree K_m, $m = 1, \ldots, M_P$ and a set $\mathcal{E}$ of edges that connect them. The edges connected to a variable node V_n or a constraint node C_m are labeled by $e_{n,j}^{\mathrm{v}}$ or $e_{m,k}^{\mathrm{c}}$, respectively, where $j = 1, \ldots, J_n$ and $k = 1, \ldots, K_m$. The j-th edge of V_n is connected to the k-th node of C_m if $e_{n,j}^{\mathrm{v}} = e_{m,k}^{\mathrm{c}}$. A protograph can be represented by means of an $M_P \times N_P$ bi-adjacency matrix $\mathbf{B}$, which is called the *base matrix* of the protograph. The entry in row m and column n of $\mathbf{B}$ is equal to the number of edges that connect nodes C_m and V_n.

Example 2: The base matrix of a regular LDPC code ensemble with variable node degree $J_n = 2$ and constraint node degree $K_n = 7$ is equal to

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \ . \tag{3}$$

The corresponding protograph is illustrated in Fig. 4. □

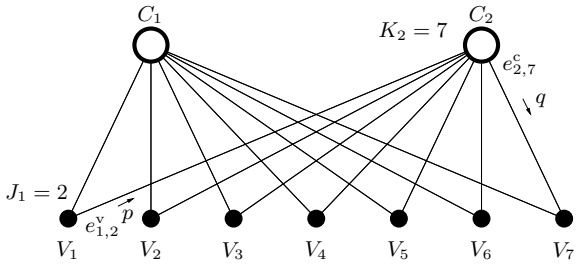

Fig. 4. Example of a regular protograph with $M_P = 2$ constraint nodes and $N_P = 7$ variable nodes.

The parity-check matrix $\mathbf{H}$ of an LDPC code can be derived from a protograph by replacing each 1 in $\mathbf{B}$ by a permutation matrix and each 0 by an all-zero matrix[1]. An ensemble of protograph based LDPC (PG-LDPC) codes of length $N = TN_P$ is defined by the set of matrices $\mathbf{H}$ that can be derived from a given protograph by all possible combinations of size T permutation matrices. Equivalently, the protograph can be interpreted as a template for the Tanner graph of a derived code, which can be obtained by a copy-and-permute operation [14]. The protograph is *lifted* by replicating each node T times and the edges are permuted among these replica in such a way that the structure of the original graph is preserved. As a consequence, a density evolution analysis for PG-LDPC ensembles can be performed within the protograph.

Assume that belief propagation is used for decoding, with log-likelihood ratios (LLRs) acting as messages. In every iteration i, first all constraint nodes and then all variable nodes are updated. The messages computed at a constraint node C_m are then equal to

$$L^{(i)}_{\mathrm{c}}(e^{\mathrm{c}}_{m,k}) = 2\mathrm{atanh}\left(\prod_{k' \neq k} \tanh\left(\frac{L^{(i-1)}_{\mathrm{v}}(e^{\mathrm{c}}_{m,k'})}{2}\right)\right) \ , \tag{4}$$

where $k, k' \in \{1, \ldots, K_m\}$. The incoming messages in the first iteration are initialized by the channel LLRs of the neighboring variable nodes, i.e., $L^{(0)}_{\mathrm{v}}(e_{m,k'}) = L_{\mathrm{ch}}(V_n)$, where V_n is the variable node connected to $e_{m,k'}$. The messages computed at a variable node V_n are equal to

$$L^{(i)}_{\mathrm{v}}(e^{\mathrm{v}}_{n,j}) = L_{\mathrm{ch}}(V_n) + \sum_{j' \neq j} L^{(i)}_{\mathrm{c}}(e^{\mathrm{c}}_{n,j'}) \ , \tag{5}$$

where $j, j' \in \{1, \ldots, J_n\}$.

For transmission over a BEC the messages that are passed between the nodes represent either an erasure or the correct symbol values 0 or 1. In this case the BP decoder is particularly simple and exact density evolution can be described explicitly. Let $q^{(i)}(e^{\mathrm{c}}_{m,k})$ denote the probability that the check to variable node message which is sent along edge $e^{\mathrm{c}}_{m,k}$ in decoding iteration i is an erasure. This is the case if at least one of the incoming messages from the other neighboring nodes is erased, i.e.,

$$q^{(i)}(e^{\mathrm{c}}_{m,k}) = 1 - \prod_{k' \neq k} \left(1 - p^{(i-1)}(e^{\mathrm{c}}_{m,k'})\right) \ , \tag{6}$$

where $p^{(i-1)}(e^{\mathrm{c}}_{m,k'})$, $k, k' \in \{1, \ldots, K_m\}$, denote the probabilities that the incoming messages computed in the previous iteration are erasures.

The variable to check node message sent along edge $e^{\mathrm{v}}_{n,j}$ is an erasure if all incoming messages from the channel and from the other neighboring check nodes are erasures. Thus we have

$$p^{(i)}(e^{\mathrm{v}}_{n,j}) = \varepsilon \prod_{j' \neq j} q^{(i)}(e^{\mathrm{v}}_{n,j'}) \ , \tag{7}$$

where $j, j' \in \{1, \ldots, J_n\}$ and ε is the erasure probability of the BEC.

Equations (6) and (7) are the well-known density evolution equations for the BEC [15] [16], applied to protograph LDPC codes [14]. They can be used to compute the density evolution *threshold* of an ensemble, defined as the maximal value of the channel parameter ε for which $p^{(i)}$ converges to zero as i tends to infinity.

B. GLDPC Codes

In protograph-based GLDPC (PG-GLDPC) codes [17], each constraint node C_m can represent an arbitrary block code $\mathcal{C}_m$ of length K_m. Assuming an LLR based BP decoder, the messages from a constraint C_m to adjacent variable nodes are then given by

$$L^{(i)}_{\mathrm{c}}(e^{\mathrm{c}}_{m,k}) = \log \sum_{\mathbf{v} \in \mathcal{C}^{m,0}_k} \prod_{k' \neq k} \exp\left(L^{(i-1)}_{\mathrm{v}}(e^{\mathrm{c}}_{m,k'})(1/2 - v_{k'})\right) - \log \sum_{\mathbf{v} \in \mathcal{C}^{m,1}_k} \prod_{k' \neq k} \exp\left(L^{(i-1)}_{\mathrm{v}}(e^{\mathrm{c}}_{m,k'})(1/2 - v_{k'})\right) \ , \tag{8}$$

where $k, k' \in \{1, \ldots, K_m\}$. Here we have partitioned $\mathcal{C}_m$ into the sets $\mathcal{C}^{m,0}_k$ and $\mathcal{C}^{m,1}_k$, corresponding to codewords $\mathbf{v}$ for which $v_k = 0$ and $v_k = 1$, respectively. The message $L^{(i)}_{\mathrm{c}}(e^{\mathrm{c}}_{m,k})$ corresponds to the k-th extrinsic output generated by an optimal APP decoder for component code $\mathcal{C}_m$, which is applied to the incoming messages of node C_m. The variable nodes are updated according to (5) like for conventional LDPC codes. Equation (8) is equivalent to (4) if $\mathcal{C}_m$ is a single parity-check code.

For a density evolution analysis of PG-GLDPC codes, equation (6) can be replaced by the general expression

$$q^{(i)}(e^{\mathrm{c}}_{m,k}) = f^{\mathcal{C}_m}_k\left(p^{(i-1)}(e^{\mathrm{c}}_{m,k'}), k' \neq k\right) \ , \tag{9}$$

where $f^{\mathcal{C}_m}_k$ is a multi-dimensional input/output transfer function that characterizes the APP decoder corresponding to (8). Note that, in general, $f^{\mathcal{C}_m}_k$ can be different for each $k \in \{1, \ldots, K_m\}$ so that the order of edges connected to node C_m can affect the performance of the ensemble. A method for computing explicit expressions for the APP decoder output distributions that can be used in (9) was presented in [7]. It is

[1]The entries of multiple edges between a pair of nodes are replaced by the sum of permutation matrices

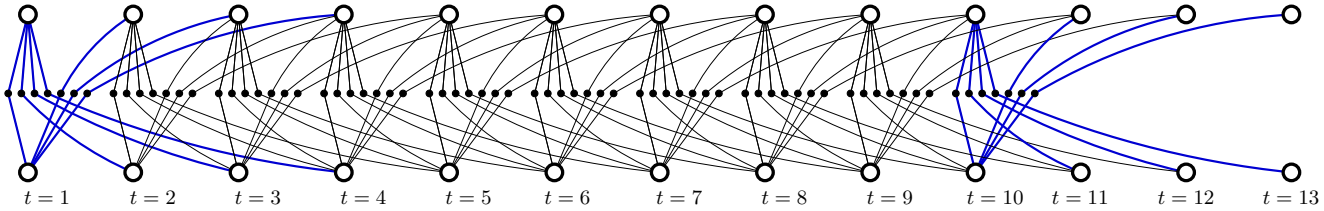

Fig. 5. Protograph of a BBC with (7,4) Hamming component codes, terminated after $L = 10$ time instants. If the symbols at $t = 1$ and $t = 10$ are perfectly known, the edges connected to these nodes can be removed and the protograph becomes equivalent to that for $L = 8$.

based on a Markov chain analysis of the decoder metrics in a trellis representation of the block code $\mathcal{C}_m$.

Example 3: Consider a length five shortened Hamming code defined by the parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} , \tag{10}$$

which can be used, e.g., as a component code of a PG-GLDPC code. For this code the extrinsic output probabilities q_n are

$$\begin{aligned} q_1 &= p_3p_5 - p_3p_4p_5p_2 + p_2p_3p_4 \\ q_2 &= p_4p_5 - p_3p_4p_5p_1 + p_1p_3p_4 \\ q_3 &= p_1p_5 - p_1p_4p_5p_2 + p_1p_2p_4 \\ q_4 &= p_2p_5 - p_2p_3p_5p_1 + p_1p_2p_3 \\ q_5 &= p_2p_4 - p_1p_2p_3p_4 + p_1p_3 \ . \end{aligned}$$

These functions, when applied in the density evolution equation (9), can be used for computing PG-GLDPC code thresholds. □

The output distributions are monotonic in their arguments: if one of the input erasure probabilities is decreased, the output probabilities stay constant or decrease as well. It follows that code shortening, which is equivalent to assigning infinite reliability to some of the symbols, reduces the output erasure probabilities, like it is the case for single parity-check equations.

IV. Analysis of terminated PG-BBCs

Taking the Tanner graph of a TBBC as a protograph we can derive ensembles of protograph based BBCs (PG-BBCs). From the array representation of a BBC with memory m_{cc} follows then a convolutional protograph with infinite base matrix

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & & \ddots & & & \\ \mathbf{B}_{m_{\text{cc}}} & \dots & \mathbf{B}_0 & & & \\ & \ddots & & \ddots & & \\ & & \mathbf{B}_{m_{\text{cc}}} & \dots & \mathbf{B}_0 & \\ & & & \ddots & & \ddots \end{bmatrix} .$$

The $M_P \times N_P$ component base matrices $\mathbf{B}_i$, $i = 0, \dots, m_{\text{cc}}$, describe the edges from the N_P variable nodes at time t to the $M_P = 2$ constraint nodes at time $t + i$.

Example 4: Consider a PG-BBC derived from a TBBC with (7,4) Hamming component codes, i.e., $N_P = 7$. The component base matrices, which follow from the graph in Fig. 3, are equal to

$$\begin{aligned} \mathbf{B}_0 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} , \\ \mathbf{B}_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} , \\ \mathbf{B}_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} , \\ \mathbf{B}_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} . \end{aligned}$$

Observe that the sum of the component base matrices is equal to the base matrix $\mathbf{B}$ of the corresponding GLDPC code, which is given in (3). This reflects the fact that the graph of the TBBC in Fig. 3 can be obtained by repeating the GLDPC graph in Fig. 4 and permuting the edges among several adjacent time instants. □

In general, the component base matrices of a PG-BBC that is derived from a TBBC by this procedure are given by

$$\mathbf{B}_0 = \begin{bmatrix} 1 & \mathbf{i} & \mathbf{0} \\ 1 & \mathbf{0} & \mathbf{i} \end{bmatrix} , \ \mathbf{B}_i = \begin{bmatrix} 0 & \mathbf{0} & \mathbf{e}_i \\ 0 & \mathbf{e}_i & \mathbf{0} \end{bmatrix} , \tag{11}$$

where $i = 1, \dots, m_{\text{cc}}$, $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ is the length m_{cc} vector with a one at the i-th position and zeros elsewhere, $\mathbf{0}$ is the all-zero vector, and $\mathbf{i}$ the all-one vector[2].

Assume now that we start encoding at time $t = 1$ and terminate after L time instants. As a result we obtain a block protograph with the $2(L + m_{\text{cc}}) \times N_P L$ base matrix

$$\mathbf{B}_{[1,L]} = \begin{bmatrix} \mathbf{B}_0 & & & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & & & \\ \vdots & & & & & & \\ \mathbf{B}_{m_{\text{cc}}} & & & & & & \\ & \ddots & & \ddots & \ddots & & \\ & & & & \mathbf{B}_1 & \mathbf{B}_0 & \\ & & & & & \vdots & \\ & & & & & \mathbf{B}_{m_{\text{cc}}} & \end{bmatrix} . \tag{12}$$

[2]In this notation we group the variable nodes in the graph slightly different from the order they are transmitted according to (1), where information bits appear first. Such a reordering has no effect on the graph structure.

For the PG-BBC in Example 4, the protograph corresponding to $\mathbf{B}_{[1,L]}$ is shown in Fig. 5 for a termination length of $L = 10$. Observe the lower degrees of constraint nodes at the start and end of the terminated protograph. These constraint nodes are now associated with shortened component codes in which the symbols of the missing edges are removed. Note that such a code shortening is equivalent to fixing these removed symbols and assigning an infinite reliability to them.

The parity-check matrix $\mathbf{H}$ of the block codes, derived from $\mathbf{B}_{[1,L]}$ by lifting with some factor T, has $N = N_P T$ columns and $M = 2M_{\mathcal{C}}T$ rows, were $M_{\mathcal{C}}$ denotes the number of parity-checks of the component code. It follows that the rate of the codes is equal to

$$R_L = 1 - \left(\frac{L+\Delta}{L}\right)\frac{2M_{\mathcal{C}}}{N_P} \tag{13}$$

for some $\Delta > 0$ that accounts for the rate loss due to the termination. As $L \to \infty$, the rate R_L converges to the rate of the corresponding regular GLDPC code. Assuming full rank of $\mathbf{H}$, the rate loss coefficient can be identified as $\Delta = m_{\mathrm{cc}}$. However, shortened component codes at the ends of the graph are causing a reduced rank of $\mathbf{H}$ that slightly increases R_L. For Hamming component codes of length $N_P = 7$ and $N_P = 15$ (with $M_{\mathcal{C}} = 3$ and $M_{\mathcal{C}} = 4$, respectively) the coefficients $\Delta = 2$ (instead of $m_{\mathrm{cc}} = 3$) and $\Delta = 5.5$ (instead of $m_{\mathrm{cc}} = 7$) could be observed by experiments with random liftings.

The terminated PG-BBCs can be interpreted as PG-GLDPC block codes of a specific structure. The length of these codes depends not only on the lifting factor T but also on the termination length L. For a fixed L, the density evolution thresholds ε_L^* corresponding to codes with base matrix $\mathbf{B}_{[1,L]}$ can be estimated by recursive application of (7) and (9) for different channel parameters ε. We computed these thresholds for Hamming component codes of length $N_P = 7$ and $N_P = 15$ and different termination lengths L. The results are given in Table I and illustrated in Figure 6, together with the Shannon limit $\varepsilon_{\mathrm{sh}} = 1 - R_L$, which follows from (13).

	$N_P = 7$	$N_P = 15$
L	ε_L^*	ε_L^*
10		0.6737
12		0.6331
15	0.8681	0.5948
17		0.5778
18	0.8584	
20	0.8567	0.5631
25		0.5429
40	0.8562	0.5287
60	0.8562	0.5279
80	0.8562	0.5278
100	0.8562	0.5278
1000	0.8562	0.5278
	ε^*	ε^*
GLDPC	0.7565	0.4678

TABLE I
THRESHOLDS OF REGULAR PG-GLDPC CODES AND OF TERMINATED PG-BBCS FOR DIFFERENT TERMINATION LENGTHS L .

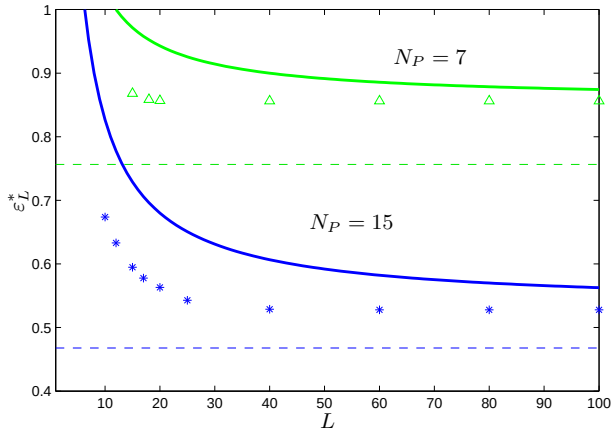

Fig. 6. Thresholds ε_L^* of terminated PG-BBCs for different termination lengths L. Hamming component codes of length $N_P = 7$ and $N_P = 15$ are considered. The solid lines show the corresponding Shannon limit $\varepsilon_{\mathrm{sh}}$. The dashed horizontal lines show the thresholds ε^* of the regular PG-GLDPC codes.

As L increases, the degree distributions approach those of the corresponding regular PG-GLDPC code with the same component codes. As expected, the thresholds of the PG-BBCs are decreasing with L. However, after a certain termination length L the thresholds approach a constant value ε_∞^*, which interestingly is still larger than the threshold ε^* of the regular PG-GLDPC code.

This phenomenon, which can also be observed for PG-LDPC CCs [9], can be explained by a closer inspection of the structure of the terminated protograph: during the iterations, the messages along edges at times $t = 1$ and $t = L$ will be the most reliable ones because the shortened component codes at the start and end of the graph provide a stronger protection. Their erasure probabilities have the potential to converge to zero even for channel parameters ε beyond the threshold of the corresponding regular GLDPC code. But when the symbols at $t = 1$ and $t = L$ are perfectly known, the connected edges can be removed from the protograph, which results in the shortened protograph $\mathbf{B}_{[2,L-1]}$, as illustrated in Fig. 5. It follows now by induction that the messages eventually converge to zero at all times $t = 1, \ldots, L$ for arbitrary values L. Theorem 1 in [9], for which a proof can be found in [18], is based on the same arguments and can be generalized to the considered PG-BBCs.

Simulation results of a PG-BBC in comparison with its GLDPC counterpart are shown in Fig. 7, together with the corresponding thresholds. In this example, a relatively large termination length $L = 1000$ was chosen to demonstrate that the performance is dominated by the special structure of the protograph and not by its degree distribution alone. Motivated by the arguments above, the decoder was operating on a sliding window to reduce the required number of updates per node. Figure 8 shows the influence of L on the performance of PG-BBCs. It is demonstrated that a short L, which causes a substantial rate reduction, can be used intentionally for improving the performance.

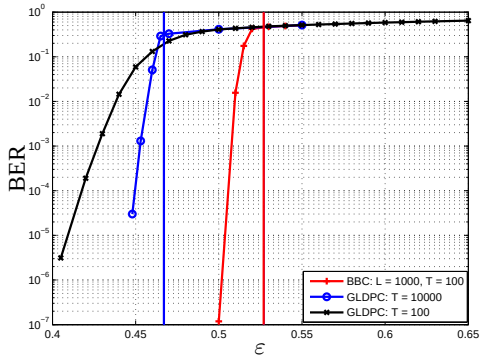

Fig. 7. Simulation results of a terminated PG-BBC with $T = 100$ and corresponding GLDPC codes with $T = 100$ and $T = 10000$. Their thresholds are shown as vertical lines for comparison. All codes are based on length $N_P = 15$ Hamming component codes.

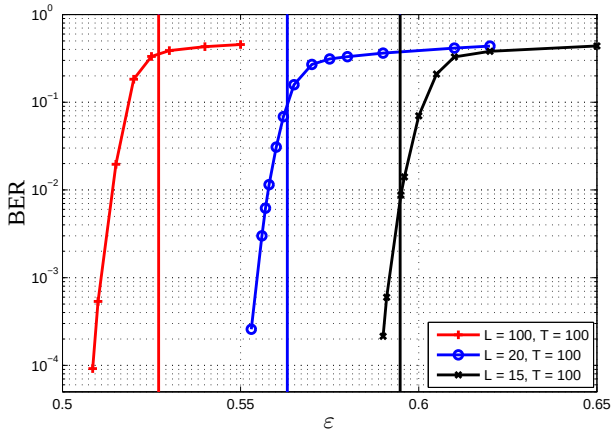

Fig. 8. Simulation results of terminated PG-BBCs with $T = 100$ for different termination lengths L. Their thresholds are shown as vertical lines for comparison. All codes are based on length $N_P = 15$ Hamming component codes.

All simulated codes were constructed using randomly chosen permutation matrices without any cycle optimization. Note that the array structure of BBCs ensures that the girth of their Tanner graph is lower bounded by eight.

V. Conclusion

We considered a class of protograph based BBCs that are constructed by lifting the Tanner graph of a TBBC. These PG-BBCs can be terminated, resulting in some specially structured PG-GLDPC block codes. Based on exact density evolution equations for PG-GLDPC codes, which can be derived for the case of the BEC, we performed a threshold analysis of the terminated PG-BBCs. It turns out that the terminated PG-BBCs have better thresholds than their regular GLDPC counterparts, even for large termination lengths where the degree distributions become nearly identical. Simulation results confirm the calculated asymptotic thresholds.

The authors are grateful for the use of the high performance computing facilities of the ZIH at TU Dresden.

References

[1] P. Elias, "Error free coding," *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 29–37, 1954, Also in *Key Papers in Development of Coding Theory*, IEEE press, New York, NY, 1974.

[2] M. Lentmaier, D.V. Truhachev, and K.Sh. Zigangirov, "Iteratively decodable sliding codes on graphs," in *Proc. of ACCT-VIII*, St. Petersburg, Russia, Sept. 2002, pp. 190–193.

[3] A.J. Feltström, D. Truhachev, M. Lentmaier, and K.Sh. Zigangirov, "Braided block codes," *IEEE Trans. Inform. Theory*, vol. 55, no. 6, pp. 2640–2658, June 2009.

[4] R. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[5] A. Jiménez Feltström and K.Sh. Zigangirov, "Periodic time-varying convolutional codes with low-density parity-check matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 2181–2190, Sept. 1999.

[6] M. Lentmaier and K. Sh. Zigangirov, "Iterative decoding of generalized low-density parity-check codes," in *Proc. IEEE International Symposium on Information Theory*, Boston, USA, August 1998, p. 149.

[7] M. Lentmaier, M.B.S. Tavares, and G.P. Fettweis, "Exact erasure channel density evolution for protograph based generalized LDPC codes," in *Proc. IEEE International Symposium on Information Theory*, Seoul, Korea, July. 2009, pp. 566–570.

[8] A. Sridharan, M. Lentmaier, D. J. Costello, Jr., and K. Sh. Zigangirov, "Convergence analysis of a class of LDPC convolutional codes for the erasure channel," in *Proceedings of the 42nd Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, 2004.

[9] M. Lentmaier, G.P. Fettweis, K.Sh. Zigangirov, and D.J. Costello, Jr., "Approaching capacity with asymptotically regular LDPC codes," in *Proc. Information Theory and Applications Workshop*, San Diego, USA, Feb. 2009.

[10] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. IT-27, no. 9, pp. 533–547, Sept. 1981.

[11] A. Sridharan, M. Lentmaier, D. V. Truhachev, D. J. Costello, Jr., and K. Sh. Zigangirov, "On the minimum distance of low-density parity-check codes with parity-check matrices constructed from permutation matrices," *Probl. Inf. Transm. (Probl. Pered. Inform.)*, vol. 41, no. 1, pp. 33–44, 2005.

[12] R.M. Tanner, D. Sridhara, A. Sridharan, T.E. Fuja, and D.J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[13] A. Sridharan, D. Sridhara, D.J. Costello, Jr., and T.E. Fuja, "A construction for irregular low density parity check convolutional codes," in *Proc. IEEE International Symposium of Information Theory*, Yokohama, Japan, June 2003, p. 4.

[14] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," in *IPN Progress Report 42-154, JPL*, Aug. 2003.

[15] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practically loss-resilient codes," in *Proc. 29th Annual ACM Symposium on Theory of Computing*, 1997, pp. 150–149.

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

[17] G. Liva, W.E. 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.

[18] M. Lentmaier, A. Sridharan, D.J. Costello, Jr., and K.Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," submitted to *IEEE Trans. Inform. Theory*.