On the Thresholds of Generalized LDPC Convolutional Codes Based on Protographs

Lentmaier, Michael; Fettweis, Gerhard

Published in:
[Host publication title missing]

DOI:
10.1109/ISIT.2010.5513595

2010

Citation for published version (APA):
Lentmaier, M., & Fettweis, G. (2010). On the Thresholds of Generalized LDPC Convolutional Codes Based on Protographs. In [Host publication title missing] (pp. 709-713). IEEE--Institute of Electrical and Electronics Engineers Inc.. DOI: 10.1109/ISIT.2010.5513595
On the Thresholds of Generalized LDPC Convolutional Codes Based on Protographs

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

Abstract—A threshold analysis of terminated generalized LDPC convolutional codes (GLDPC CCs) is presented for the binary erasure channel. Different ensembles of protograph-based GLDPC CCs are considered, including braided block codes (BBCs). It is shown that the terminated PG-GLDPC CCs have better thresholds than their block code counterparts. Surprisingly, our numerical analysis suggests that for large termination factors the belief propagation decoding thresholds of PG-GLDPC CCs coincide with the ML decoding thresholds of the corresponding PG-GLDPC block codes.

I. INTRODUCTION

It can be observed that terminated protograph-based LDPC convolutional codes (PG-LDPC CCs) have better belief propagation (BP) decoding thresholds than their tailbiting versions or the block codes their are constructed from [1]. An analysis reveals that this can be prescribed to the slight irregularity at the ends of their Tanner graphs. This effect is visible even if the termination factor tends to infinity and both the code rate and degree distributions approach those of the corresponding block codes. A comparison with upper bounds on ML decoding thresholds presented in [2] shows that for infinite termination factors the BP thresholds ε∞,BP of regular LDPC CC ensembles must be close to the ML decoding thresholds of the regular block codes for both the binary erasure channel (BEC) and the additive white Gaussian noise (AWGN) channel. It can be observed, that on the BEC they actually numerically coincide with the tighter upper bounds εblk,ML on the ML thresholds presented in [3], which are based on BP EXIT functions [4] (see Table I). The same can also be verified for the convolutional code version of the irregular ARJA ensembles [5] investigated in [6]. More recently, for regular codes and the BEC, the equality of BP thresholds of convolutional ensembles and ML thresholds of the underlying block ensembles has been proven analytically in [7].

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
 Ensemble & εblk,ML & ε∞,BP & εblk,ML \\
\hline
 (3,6) & 0.4294 & 0.4881 & 0.4881 \\
 (4,8) & 0.3834 & 0.4977 & 0.4977 \\
 (5,10) & 0.3416 & 0.4994 & 0.4994 \\
 ARJA & 0.4387 & 0.4997 & 0.4997 \\
\hline
\end{tabular}
\caption{Thresholds of some PG-LDPC ensembles.}
\end{table}

In this paper, based on the transfer functions derived in [8], we extend the threshold analysis of PG-LDPC CCs [1] [6] to generalized LDPC (GLDPC) codes. In addition to the recently investigated protograph-based braided block codes (PG-BBCs) [10] [11] we present some further PG-GLDPC ensembles and compare their thresholds with the upper bounds εblk,ML. It turns out that ε∞,BP and εblk,ML are numerically equal for all investigated PG-GLDPC ensembles.

II. THRESHOLDS OF PROTOGRAPH ENSEMBLES

A. Protograph GLDPC Codes

A protograph is a bipartite graph consisting of a set of variable nodes Vn with degree Jn, n = 1,...,NP, a set of constraint nodes Cm with degree Kn, m = 1,...,MP and a set of edges that connect them. The edges connected to a variable node Vn or a constraint node Cm are labeled by εn,j or εm,k, respectively, where j = 1,...,Jn and k = 1,...,Kn. The j-th edge of Vn is connected to the k-th node of Cm if εn,j = εm,k. In protograph-based GLDPC (PG-GLDPC) codes, each constraint node Cm can represent an arbitrary block code Cm of length Km. As an example, Fig. 1 shows a rate R = 1/6 "Hamming-doped" protograph [12], created from an accumulate-repeat-accumulate (ARA) code. This ARA protograph was "doped" by associating a shortened (6,3) Hamming code to one of its check nodes.

A protograph can be represented by means of an MP × NP bi-adjacency matrix B, which is called the base matrix of the protograph. The entry in row m and column n of B is equal to the number of edges that connect nodes Cm and Vn. The
where \( j, j' \) particularly simple and exact density evolution can be described

**B. Density Evolution for PG-GLDPC Codes**

In order to construct ensembles of PG-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 operation [13]. 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-GLDPC ensembles can be performed within the protograph. Equivalently, the Tanner graph of an GLDPC code can be represented by a bi-adjacency matrix that is derived from a protograph by replacing each 1 in \( B \) by a permutation matrix and each 0 by an all-zero matrix \(^1\). An ensemble of protograph based GLDPC (PG-GLDPC) codes of length \( N = TN_P \) is defined by the set of matrices that can be derived from a given protograph by all possible combinations of size \( T \) permutation matrices.

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

\[
L_c^{(i)}(\epsilon^{c}_{m,k}) = \log \sum_{v \in C^{m,0}_k \cup C^{m,1}_k} \prod_{k' \neq k} \exp \left( L^{(i-1)}_v(\epsilon^{c}_{m,k'}) (1/2 - v_{k'}) \right) - \log \sum_{v \in C^{m,1}_k} \prod_{k' \neq k} \exp \left( L^{(i-1)}_v(\epsilon^{c}_{m,k'}) (1/2 - v_{k'}) \right),
\]

where \( k, k' \in \{1, \ldots, K_m\} \). Here we have partitioned \( C_m \) into the sets \( C^{m,0}_k \) and \( C^{m,1}_k \), corresponding to codewords \( v \) for which \( v_k = 0 \) and \( v_k = 1 \), respectively. The message \( L_c^{(i)}(\epsilon^{c}_{m,k}) \) corresponds to the \( k \)-th extrinsic output generated by an optimal APP decoder for component code \( C_m \), which is applied to the incoming messages of node \( C_m \). The incoming messages in the first iteration are initialized by the channel LLRs of the neighboring variable nodes, i.e., \( L_v^{(0)}(\epsilon^{c}_{m,k}) = L_{ch}(V_n) \), where \( V_n \) is the variable node connected to \( c_{m,k} \).

The messages computed at a variable node \( V_n \) are equal to

\[
L_v^{(i)}(\epsilon^{v}_{n,j}) = L_{ch}(V_n) + \sum_{j' \neq j} L_c^{(i)}(\epsilon^{c}_{n,j'})
\]

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

**B. Density Evolution for PG-GLDPC Codes**

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 \( q^{(i)}(\epsilon^{c}_{m,k}) \) denote the probability that the check to variable node message which is sent along edge \( c_{m,k} \) in decoding iteration \( i \) is an erasure. Assuming a conventional LDPC code, where \( C_m \) is a single parity-check code, this is the case if at least one of the incoming messages from the other neighboring nodes is erased, i.e.,

\[
q^{(i)}(\epsilon^{c}_{m,k}) = 1 - \prod_{k' \neq k} \left( 1 - p^{(i-1)}(\epsilon^{c}_{m,k'}) \right),
\]

where \( p^{(i-1)}(\epsilon^{c}_{m,k'}), k, k' \in \{1, \ldots, K_m\} \), denote the probabilities that the incoming messages computed in the previous iteration are erasures. In case of an arbitrary block code \( C_m \), equation (4) can be replaced by the general expression

\[
q^{(i)}(\epsilon^{c}_{m,k}) = f_k^{c_m} \left( p^{(i-1)}(\epsilon^{c}_{m,k'}); k' \neq k \right),
\]

where \( f_k^{c_m} \) is a multi-dimensional input/output transfer function that characterizes the APP decoder that computes the messages \( L^{(i)}_v(\epsilon^{c}_{m,k}) \) corresponding to (2). Note that, in general, \( f_k^{c_m} \) 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 (5) was presented in [8]. It is based on a Markov chain analysis of the decoder metrics in a trellis representation of the block code \( C_m \).

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

\[
p^{(i)}(\epsilon^{v}_{n,j}) = \varepsilon \prod_{j' \neq j} q^{(i)}(\epsilon^{v}_{n,j'})
\]

where \( j, j' \in \{1, \ldots, J_n\} \) and \( \varepsilon \) is the erasure probability of the BEC.

**C. Upper-bounding ML Thresholds with BP EXIT Functions**

The extrinsic probability \( p_{BP,\text{extr}} \), that a symbol associated with variable node \( V_n \) remains erased after BP decoding with \( I \) iterations can be expressed as

\[
p_{BP,\text{extr}}(V_n, \varepsilon) = \prod_{j} q^{(I)}(\epsilon^{v}_{n,j})
\]

Note that here the product is over all incoming messages of \( V_n \) and the intrinsic channel erasure probability is omitted in the expression but implicitly involved in the calculation of \( q^{(I)}(\epsilon^{v}_{n,j}) \). The BP EXIT function \( h_{BP}(\varepsilon) \) [4] is given by the average of \( p_{BP,\text{extr}} \) over all transmitted variable nodes \(^2\).

Consider a (2,7) regular PG-GLDPC ensemble with Hamming component codes of length \( N_{cc} = 7 \), as illustrated in Fig. 2. The BP EXIT function of this ensemble is shown in Fig. 3. The vertical line indicates the channel value at which the grey area below the curve is equal to the rate of the ensemble, which forms an upper bound \( \varepsilon_{ML} \) on the threshold of an optimal ML decoder. This follows from the area theorem [14] and the fact that \( h_{BP}(\varepsilon) \) can never be below \(^2\)

\(^1\)The entries of multiple edges between a pair of nodes are replaced by the sum of permutation matrices

\(^2\)Punctured nodes which are not transmitted are excluded.
III. TERMINATED PG-GLDPC CONVOLUTIONAL CODES

Analogously to block codes, an ensemble of GLDPC convolutional codes can be constructed from a protograph. Such protograph-based GLDPC convolutional codes (PG-GLDPC convolutional codes can be constructed from a protograph. Such PG-BBCs based on component codes of length $N_{cc}$ can be obtained by repeating the GLDPC graph in Fig. 2 and permuting the edges among several adjacent time instants. This reflects the fact that the graph of the PG-BBC in Fig. 4 corresponds to the possible assignment of edges, resulting in a convolutional protograph with the same variable and check node degrees as the original block protograph.

A. Protograph-Based Braided Block Codes

The protograph-based BBC (PG-BBC) ensembles considered in [11] are an example of PG-GLDPC CC ensembles as defined above. These can be derived by using the Tanner graph of a tightly BBC [10] as a protograph. The component base matrices of such a PG-BBC can be identified as

$$B_0 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & i \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 & 0 & e_i \\ 0 & e_i & 0 \end{bmatrix},$$

where $i = 1, \ldots, m_{cc}$, $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the length $m_{cc}$ vector with a one at the $i$-th position and zeros elsewhere, $0$ is the all-zero vector, and $i$ the all-one vector. Throughout this paper we use the term Ensemble $A_{N_{cc}}$ when referring to such PG-BBCs based on component codes of length $N_{cc}$.

For PG-BBCs with (7,4) Hamming component codes (i.e., $N_{cc} = 7$ and $M_{cc} = 2$), the resulting convolutional protograph is illustrated in Fig. 4. Its girth is equal to eight, which follows from the structure of the array and is true for any tightly BBC. Observe that the sum of the component base matrices is equal to the base matrix $B$ of the corresponding GLDPC code, which is the all-one matrix of dimension $M_{cc} \times N_{cc} = 2 \times 7$. This reflects the fact that the graph of the PG-BBBC in Fig. 4 can be obtained by repeating the GLDPC graph in Fig. 2 and permuting the edges among several adjacent time instants.

B. Convolutional Protographs with $m_{cc} = 1$

Following the edge-spreading approach, considered in [6] for the design of PG-LDPC CCs, let us now consider some alternative constructions of PG-GLDPC CC ensembles. Starting from an arbitrary block protograph, defined by an $M_{cc} \times N_{cc}$ base matrix $B$, we divide the edges among times $t, t+1, \ldots, t+m_{cc}$. For a given target memory $m_{cc}$, any set of component base matrices $B_0, B_1, \ldots, B_{m_{cc}}$ which satisfies the condition

$$\sum_{i=0}^{m_{cc}} B_i = B$$

(9)

corresponds to a possible assignment of edges, resulting in a convolutional protograph with the same variable and check node degrees as the original block protograph.

For the case $N_{cc} = 7$, consider splitting $B$ into component base matrices

$$B_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$
resulting in an ensemble of PG-GLDPC CCs with memory $m_{cc} = 1$, which we call Ensemble $B_L$. A segment of the corresponding convolutional protograph is shown in Fig. 5(left). A generalization to the Ensemble $B_{15}$ and other values of $N_{cc}$ is straightforward. While we can see that the reduced memory leads to a more compact representation of PG-GLDPC CCs, there is still some sparsity observable, reflected in a relatively low fraction of non-zero elements in the component base matrices $B_0$ and $B_1$. In case of even $N_{cc}$ this can be eliminated by introducing multiple edges in the base matrix $B$ of the block protograph. Considering shortened $(14,10)$ Hamming codes as an example, we can use the nodes from the $2 \times 7$ protograph in Fig. 4 and connect each variable/check node pair with a double edge. We split the corresponding base matrix

$$B_0 = B_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (10)$$

and obtain the protograph of Ensemble $C_{14}$ with $M_P = 1$ check nodes and $N_P = 7$ variable nodes at each time instant, a segment of which is illustrated in Fig. 5(right). Since now $N_P = 7 = N_{cc}/2$, the ensemble has only half the constraint length as Ensemble $B_{14}$ for a given lifting factor $L_T$. Puncturing the first variable node at each time instant $t$ results in Ensemble $C_{14,p}$ with the asymptotic design rate $R_{cc} = 0.5$.

C. Thresholds of Terminated PG-GLDPC CCs

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 base matrix

$$B_{[1,L]} = \begin{bmatrix} B_0 & \cdots & \cdots & B_0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ B_{m_{cc}} & \cdots & \cdots & B_{m_{cc}} \end{bmatrix}_{M_P (L+m_{cc}) \times N_P L}. \quad (11)$$

This terminated protograph is slightly irregular with lower constraint node degrees at the start and end. These shortened 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 variable node degrees are not affected by termination.

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

$$R_L = 1 - \left( \frac{L + \Delta}{L} \right) \frac{M_P M_C}{N_P} \quad (12)$$

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 $H$, the rate loss coefficient can be identified as $\Delta = m_{cc}$. However, shortened component codes at the ends of the graph can cause a reduced rank of $H$ that slightly increases $R_L$. In our examples, a rank loss could be observed for the PG-BBC Ensembles $A_7$ and $A_{15}$, based on Hamming component codes of length $N_P = 7$ and $N_P = 15$ (with $M_C = 3$ and $M_C = 4$, respectively), where the coefficients $\Delta = 2$ (instead of $m_{cc} = 3$) and $\Delta = 5.5$ (instead of $m_{cc} = 7$) could be observed by experiments with random liftings. No rank loss was observed for the considered ensembles with $m_{cc} = 1$.

The terminated PG-GLDPC CCs can be interpreted as PG-GLDPC block codes that inherit the structure of the convolutional codes. The length of these codes depends not only on the lifting factor $T$ but also on the termination factor $L$. For a fixed $L$, the density evolution thresholds $\varepsilon_{L,BP}$ corresponding to codes with base matrix $B_{[1,L]}$ can be estimated by recursive application of (6) and (5) for different channel parameters $\varepsilon$. The resulting thresholds for ensembles $B_7$ and $B_{15}$ are compared in Fig. 6 with the BBC ensembles $A_7$ and $A_{15}$ for different termination factors $L$. The thresholds of all the considered ensembles versus the code rate are shown in Fig. 7. Analogously to PG-LDPC CCs (see Table I), we observe that as $L \to \infty$ the BP thresholds numerically coincide with the upper bounds on the ML decoding thresholds of the corresponding block code ensembles. From the achievability of $\varepsilon_{cc,BP}$ it follows that the bounds $\varepsilon_{cc,ML}$ are tight.
The BP EXIT functions of the terminated codes from ensemble $B_7$ are shown in Fig. 8. Indeed, with increasing $L$ their BP thresholds converge to their ML thresholds, indicating optimality of BP decoding. Large $L$ can be realistic in conjunction with window based decoders, like suggested in [15], where decoding delay and storage requirement depends on the window size $W$, where $W < L$, but is totally independent of the length $L$ of the transmitted code sequences. For shorter $L$, which introduces a significant rate loss, the BP decoding of the terminated codes is clearly suboptimal but still provides a flexible adjustment of coding rate and threshold.

ACKNOWLEDGMENT

The authors would like to thank Dr. Gianluigi Liva for pointing out the numerical equality of the ML thresholds of regular block codes and the BP thresholds of asymptotically regular convolutional codes. The authors are also grateful for the use of the high performance computing facilities of the ZIH at TU Dresden.

REFERENCES


