Exact Erasure Channel Density Evolution for Protograph-Based Generalized LDPC Codes

Lentmaier, Michael; Tavares, Marcos B.S.; Fettweis, Gerhard

Published in:
2009 IEEE International Symposium on Information Theory

DOI:
10.1109/ISIT.2009.5205688

2009

Exact Erasure Channel Density Evolution for Protograph-Based Generalized LDPC Codes

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

***Abstract*—We derive explicit density evolution equations for protograph-based generalized LDPC codes on the binary erasure channel. They are obtained from an analysis of multi-dimensional input/output transfer functions of the component decoders. Belief propagation decoding with optimal component APP decoders is considered. Based on the resulting transfer functions, a threshold analysis is performed for some protograph examples.**

I. Introduction

The binary erasure channel (BEC) is often used as a model for theoretical analysis of iterative decoding. In fact, the notion of irregular LDPC code ensembles, which nowadays is widely used in the design of capacity approaching codes, has been first introduced for the BEC [1] [2]. For this channel model, the density evolution equations for LDPC code ensembles can be described analytically, both for random irregular ensembles [2] and structured protograph ensembles [3].

Protographs with identical degree distributions can have different thresholds, and a careful protograph selection can lead to better thresholds compared to corresponding unstructured random ensembles. An example where structure, imposed in a controlled way, can improve performance are graphs containing variable nodes of degree one [4]. In an unstructured ensemble such nodes usually affect the decoder convergence. Another example are protograph-based terminated LDPC convolutional codes [5], which can have thresholds close to capacity with degree distributions approaching those of regular LDPC codes. An additional advantage of protographs is that they are well suited for the design of quasi-cyclic codes. Also generalized LDPC (GLDPC) codes based on protographs, where the check equations are replaced by stronger block code constraints [6], have been considered in the literature [7] [8].

In this paper, we analyze iterative decoding of protograph GLDPC codes and conceptually describe the corresponding density evolution equations for the BEC, which can be applied for determining the asymptotic convergence thresholds. These density evolution equations rely on explicit multi-dimensional input/output transfer functions of the *a posteriori* probability (APP) decoders that are applied within belief propagation (BP) to the component codes associated with the constraint nodes of the graph. Our main contribution is an analysis of these decoders, which results in a method to compute exact analytic expressions for the density evolution equations of protograph GLDPC codes. For demonstration purposes, we have applied these equations to some examples of protograph ensembles and present the corresponding thresholds.

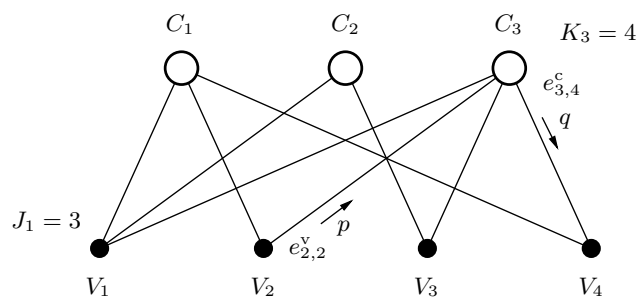

Fig. 1. Example of a protograph with $M_c = 3$ constraint nodes and $N_v = 4$ variable nodes.

II. Density Evolution on Protographs

A protograph [3] is a bipartite graph consisting of a set of variable nodes V_n with degree J_n, $n = 1, \ldots, N_v$, a set of constraint nodes C_m with degree K_m, $m = 1, \ldots, M_c$ 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 $e^{\mathrm{v}}_{n,j}$ or $e^{\mathrm{c}}_{m,k}$, 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 $e^{\mathrm{v}}_{n,j} = e^{\mathrm{c}}_{m,k}$. This way of labeling takes into account the order of edges connected to a node and allows to distinguish multiple edges between a pair of nodes.

While a protograph is formally equivalent to a Tanner graph [6], it actually represents a family of codes of different lengths whose individual Tanner graphs are obtained from the protograph by a copy-and-permute operation [3]. Then a size T permutation matrix is associated with each edge in the protograph and each node is replicated T times, resulting in a *derived graph* that defines a code of length TN_v. By this procedure, 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 the resulting codes can be performed within the protograph.

In protograph-based GLDPC codes [7] [8], each constraint node C_m can represent an arbitrary block code $\mathcal{C}_m$ of length K_m. We assume that belief propagation is used for the decoding, after transmission over a BEC with erasure probability ε. In every iteration, first all constraint nodes and then all variable

nodes are updated. The messages that are passed between the nodes represent either an erasure or the correct symbol values 0 or 1. Let $q^{(i)}(e^{\mathrm{c}}_{m,k})$ denote the probability that the constraint to variable node message which is sent along edge $e^{\mathrm{c}}_{m,k}$ in decoding iteration i is an erasure. In case of BP decoding, this message is equal to the k-th extrinsic output generated by an APP decoder of the constraint code $\mathcal{C}_m$. Then

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

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

The variable to constraint 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 constraint nodes are erasures. Thus we have

$$p^{(i)}(e^{\mathrm{v}}_{n,j}) = \varepsilon \prod_{j' \neq j} q^{(i)}(e^{\mathrm{v}}_{n,j'}) \quad , \tag{2}$$

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

In a conventional protograph LDPC code, where all constraint nodes represent a single parity-check (SPC) equation, the constraint to variable node message is an erasure if at least one of the incoming messages from the other neighboring nodes is erased. In this case (1) becomes

$$q^{(i)}(e^{\mathrm{c}}_{m,k}) = 1 - \prod_{k' \neq k}\left(1 - p^{(i-1)}(e^{\mathrm{c}}_{m,k'})\right) \quad . \tag{3}$$

Equations (3) and (2) are the well-known density evolution equations for the BEC [1] [2], applied to protograph LDPC codes [3]. For the random irregular code ensembles analyzed in [2], the degrees of constraint nodes and variable nodes are random variables and density evolution tracks the *average* message distributions over all codes in an ensemble, which are equal along all edges in the graph. A protograph, on the other hand, imposes a deterministic structure on the neighborhood of each node so that the message distributions in (3) and (2) behave differently and have to be tracked separately for each edge of the protograph.

A corresponding threshold analysis for protograph GLDPC codes can be performed by means of the density evolution equations (1) and (2). Note that, in general, $f_k^{\mathcal{C}_m}$ 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. In the following section we present a simple method for computing explicit expressions for the APP decoder output distributions that can be used in (1).

III. Extrinsic Output Erasure Probabilities of APP Decoded Linear Block Codes

It has been observed in [9] that on the BEC the EXIT functions of a block code can be related to its *support weights* and *information functions* [10]. A refinement of these, the *split information functions*, which have been introduced in [9] for an EXIT chart analysis of GLDPC ensembles, allow the partition of the code into groups of symbols with different input erasure probabilities. The split information functions have also been used in [11] to derive a stability condition for random irregular doubly generalized LDPC ensembles. On the BEC there is a one-to-one correspondence between the APP decoder output erasure probabilities and the EXIT functions, namely $I_E(p) = 1 - q(p)$ and $I_A = 1 - p$. However, in the context of [9] and [11], in order to arrive at a one-dimensional chart, the *average* input and output erasure probabilities are considered. In a protograph analysis, on the other hand, the *distinct* message distributions of the different input and output symbols have to be considered, as they appear in (1) and (2). This means that multi-dimensional EXIT functions are required, which are difficult to visualize in a chart.

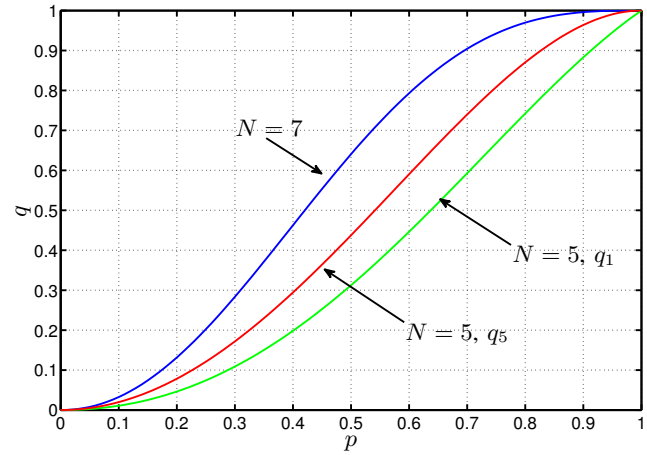

Fig. 2. The output probabilities of a (7,4) Hamming code and its shortened (5,2) code (see Example 1).

Example 1: The extrinsic output probabilities q_n of a (7,4) Hamming code with equal input probabilities $p_i = p$, $i = 1, \ldots, 7$ is given by $q_n = 3p^2 + 4p^3 - 15p^4 + 12p^5 - 3p^6$ for all $n = 1, \ldots, 7$. A plot of this function is shown in Fig. 2. If the code is shortened to length five by removing the first two symbols from each codeword, it turns out that (the derivation will follow below)

$$\begin{aligned} q_1 =& p_3p_5 - p_3p_4p_5p_2 + p_2p_3p_4 \\ q_2 =& p_4p_5 - p_3p_4p_5p_1 + p_1p_3p_4 \\ q_3 =& p_1p_5 - p_1p_4p_5p_2 + p_1p_2p_4 \\ q_4 =& p_2p_5 - p_2p_3p_5p_1 + p_1p_2p_3 \\ q_5 =& p_2p_4 - p_1p_2p_3p_4 + p_1p_3 \end{aligned} \tag{4}$$

Equivalently, we could use the corresponding functions of the (7,4) Hamming code and set the first two input probabilities to zero (infinite reliability). If we now consider equal input probabilities p, then we obtain the probabilities

$$\begin{aligned} q_i =& p^2 + p^3 - p^4 \quad , \quad i = 1, \ldots, 4 \\ q_5 =& 2p^2 - p^4 \quad , \end{aligned}$$

which are also shown in Fig. 2. This illustrates that even

for *averaged* input probabilities the output probabilities, in general, need not be identical for different code symbols. We can also see that code shortening reduces the output erasure probabilities, like it is the case for single parity-check equations. □

For the derivation of the multi-dimensional transfer function of the APP decoder, instead of redefining the support weights and information functions for multiple edge types, we use in this paper a trellis representation of block codes and perform a Markov chain analysis of the decoder metrics. For convolutional codes, this approach has been introduced for the analysis of Viterbi decoding on the BSC [12] [13]. A generalization to the analysis of APP decoding on the BEC has been presented in [14]. For the analysis of block codes, we consider the Johansson-Zigangirov (JZ) APP algorithm [15], which performs a single forward recursion through the syndrome trellis of the code.

A. Forward APP Decoding with the JZ Algorithm

Consider the decoding of some linear binary block code $\mathcal{C}$ of length N. Assume that a codeword $\boldsymbol{v} \in \mathcal{C}$ has been transmitted and some soft information is available in terms of the likelihood ratio vector $\boldsymbol{\Lambda}$. If $\boldsymbol{v}$ is transmitted over a memoryless channel, the elements of $\boldsymbol{\Lambda}$ can be written as

$$\Lambda_n = \frac{p(r_n|v_n=0)}{p(r_n|v_n=1)} \cdot \frac{p(v_n=0)}{p(v_n=1)} \tag{5}$$

for $n \in \mathcal{N} = \{1,\dots,N\}$, where $\boldsymbol{r}$ is a vector containing the received channel output values. The second term in (5) corresponds to a-priori information that may be available for the individual code symbols. In the considered iterative GLDPC decoder, the values Λ_n are given by the variable to constraint node messages.

An optimal APP decoder computes the extrinsic likelihood ratios

$$\Lambda_n^{\mathrm{e}} = \frac{\sum_{\boldsymbol{v}\in\mathcal{C}_n^0} \prod_{i\in\mathcal{N}_{\setminus n}} \gamma_i(v_i)}{\sum_{\boldsymbol{v}\in\mathcal{C}_n^1} \prod_{i\in\mathcal{N}_{\setminus n}} \gamma_i(v_i)} \ , \quad \gamma_n(v_n) = \Lambda_n^{1/2-v_n} \tag{6}$$

which can be combined with Λ_n to the a-posteriori likelihood ratios

$$\Lambda_n \cdot \Lambda_n^{\mathrm{e}} = \frac{P(v_n=0|\boldsymbol{\Lambda},\boldsymbol{v}\in\mathcal{C})}{P(v_n=1|\boldsymbol{\Lambda},\boldsymbol{v}\in\mathcal{C})} \ . \tag{7}$$

In (6) we have partitioned the code into the sets $\mathcal{C}_n^0$ and $\mathcal{C}_n^1$ consisting of all codewords for which v_n is fixed to zero and one, respectively. In the set $\mathcal{N}_{\setminus n}$, position n is excluded, i.e., $\mathcal{N}_{\setminus n} = \{1,\dots,N\} \setminus n$.

Consider now the coset $\mathcal{C}^{(n)} = \{\boldsymbol{v}_n | \boldsymbol{v}_n = \boldsymbol{v} + \boldsymbol{\epsilon}_n, \boldsymbol{v}\in\mathcal{C}\}$, where $\boldsymbol{\epsilon}_n = (0,\dots,0,1,0,\dots,0)$ is a vector with a one at the n–th position and zeros elsewhere. We introduce the values

$$\mu_0 = \sum_{\boldsymbol{v}\in\mathcal{C}} \prod_{i\in\mathcal{N}} \gamma_i(v_i) \tag{8}$$

$$\mu_n = \sum_{\boldsymbol{v}\in\mathcal{C}^{(n)}} \prod_{i\in\mathcal{N}} \gamma_i(v_i) \ , \quad n\in\mathcal{N} \ . \tag{9}$$

It can be shown [15] [16] that the decoder outputs given in (6) can be expressed as

$$\Lambda_n^{\mathrm{e}} = \frac{\mu_0 \cdot \Lambda_n - \mu_n}{\mu_n \cdot \Lambda_n - \mu_0} \ . \tag{10}$$

The JZ algorithm makes use of the fact that μ_0 and μ_n can be computed efficiently in a single forward recursion through the unexpurgated syndrome trellis of the code. Let $\mu(\boldsymbol{s},n)$ denote the metric of node $\boldsymbol{s}\in\mathbb{F}_2^{N-K}$ at level $n\in\mathcal{N}$ of the trellis. For a given state $\boldsymbol{s}$ at level $n-1$ and code symbol v_n, the state $\boldsymbol{s}'$ at level n is equal to $\boldsymbol{s}' = \boldsymbol{s} + v_n \cdot \boldsymbol{h}_n$, where $\boldsymbol{h}_n$ denotes the n-th column of a given parity-check matrix $\boldsymbol{H}$ defining $\mathcal{C}$. This leads to the recursion

$$\begin{aligned} &\mu(\boldsymbol{0},0) = 1 \ , \quad \mu(\boldsymbol{s},0) = 0 \ , \quad \forall \boldsymbol{s} \neq \boldsymbol{0} \\ &\mu(\boldsymbol{s},n) = \mu(\boldsymbol{s},n-1)\cdot\gamma_n(v_n=0) \\ &\qquad + \mu(\boldsymbol{s}+\boldsymbol{h}_n,n-1)\cdot\gamma_n(v_n=1) \ . \end{aligned} \tag{11}$$

Since all vectors in a coset $\mathcal{C}^{(n)}$ have syndrome $\boldsymbol{h}_n$, it follows from the definition of the syndrome former trellis that

$$\begin{aligned} \mu_0 &= \mu(\boldsymbol{0},N) \\ \mu_n &= \mu(\boldsymbol{h}_n,N) \ , n\in\mathcal{N} \ , \end{aligned}$$

and the extrinsic decoder outputs can be obtained from (10).

B. Computation of the Output Distributions

At each level $n = 0,\dots,N$ in the trellis there are at most $S = 2^{N-K}$ nodes that correspond to reachable states $\boldsymbol{s}$. We introduce the metric vectors $\boldsymbol{\mu}_n = (\mu(\boldsymbol{s}_1,n),\mu(\boldsymbol{s}_2,n),\dots,\mu(\boldsymbol{s}_S,n))$, where $\boldsymbol{s}_i$, $i = 1,\dots,S$ denote the different possible trellis states. The values $\mu(\boldsymbol{s}_i,n)$ depend on the parity-check matrix $\boldsymbol{H}$ and on the decoder input $\boldsymbol{\Lambda}$. Let $\mathcal{M}_n = \{\boldsymbol{\sigma}_n^{(1)},\boldsymbol{\sigma}_n^{(2)},\dots\}$ denote the set of possible metric vectors $\boldsymbol{\mu}_n$ corresponding to all the different input vectors $\boldsymbol{\Lambda}$. In case of the BEC, we have $\Lambda_n = \infty$ and $\Lambda_n = 0$ if the input message is equal to zero and one, respectively, and $\Lambda_n = 1$ if the input message is an erasure. Due to linearity of the code, we can assume in the analysis that the all-zero codeword has been transmitted. Since the nonzero elements of a metric vector $\boldsymbol{\mu}_n$ are always equal, we can normalize these entries to 1.

Example 2: Consider a length five shortened Hamming code defined by

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

The decoder starts in the state $\boldsymbol{s}_0$ and can reach the states $\boldsymbol{s}_0$ and $\boldsymbol{s}_5$, from which it then additionally can reach states $\boldsymbol{s}_3$ and $\boldsymbol{s}_6$. The possible metric vectors at levels one and two are given by

$$\begin{aligned} \boldsymbol{\mu}_1 &\in \{(1,0,0,0,0,0,0,0),(1,0,0,0,0,1,0,0)\} \\ \boldsymbol{\mu}_2 &\in \{(1,0,0,0,0,0,0,0),(1,0,0,1,0,0,0,0), \\ &\qquad (1,0,0,0,0,1,0,0),(1,0,0,1,0,1,1,0)\} \end{aligned}$$

At levels three, four, and five there exist 8, 12, and 13 different metric vectors, respectively. □

Our analysis of the decoder is based on two key properties:

1) the number of different metric vectors $|\mathcal{M}_n|$ at each level $n \in \mathcal{N}$ is finite
2) the set $\mathcal{M}_n$ is independent of the distribution $P(\boldsymbol{\Lambda})$ of the decoder input

The first property allows us to recursively calculate the distributions $P(\boldsymbol{\mu}_n)$ of the metric vectors at the different levels of the trellis for a given input distribution. This is also possible for other discrete channels, e.g., the BSC. The second property is specific for the BEC and allows us to derive explicit formulas for $P(\boldsymbol{\mu}_n)$ as function of the input erasure probabilities $p_i = P(\Lambda_i = 1)$, $i = 1, \ldots, n$.

The sequence $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N$ of metric vectors forms a Markov chain with varying transition matrices $\boldsymbol{M}_n$ of dimension $|\mathcal{M}_{n-1}| \times |\mathcal{M}_n|$. The first vector $\boldsymbol{\mu}_0$ is unique, i.e., $|\mathcal{M}_0| = 1$, since decoding starts in the all-zero state $\boldsymbol{s} = \boldsymbol{0}$ with fixed metric $\mu(\boldsymbol{0}, 0) = 1$. The element in row j and column k of $\boldsymbol{M}_n$ is equal to the probability to come from state $\boldsymbol{\mu}_{n-1} = \boldsymbol{\sigma}_{n-1}^{(j)}$ to state $\boldsymbol{\mu}_n = \boldsymbol{\sigma}_n^{(k)}$, which depends on the input erasure probability p_n. The distribution $P(\boldsymbol{\mu}_N)$ of the metric vector $\boldsymbol{\mu}_N$ at the last trellis level can be computed by

$$\left[P(\boldsymbol{\mu}_N = \boldsymbol{\sigma}_N^{(1)}) \quad P(\boldsymbol{\mu}_N = \boldsymbol{\sigma}_N^{(2)}) \quad \cdots\right] = \prod_{n=1}^{N} \boldsymbol{M}_n \ . \quad (13)$$

Our goal is to compute the extrinsic output erasure probabilities $q_n = P(\Lambda_n^{\mathrm{e}} = 1)$. From (10) we see that $\Lambda_n^{\mathrm{e}} = 1$ is satisfied for all values of Λ_n when μ_0 is equal to μ_n. It follows that

$$q_n = \sum_{\boldsymbol{\sigma} \in \mathcal{M}_n^{\mathrm{e}}} P(\boldsymbol{\mu}_N = \boldsymbol{\sigma} | p_n = 0) \quad (14)$$

where $\mathcal{M}_n^{\mathrm{e}} = \{\boldsymbol{\mu}_N : \mu(\boldsymbol{h}_n, N) = \mu(\boldsymbol{0}, N)\}$. Since we are interested in the extrinsic probabilities, the terms in (14) are conditioned on the event $p_n = 0$. They follow from (13) by substitution.

Example 3: For the code considered in Example 2, the first two transition matrices are given by

$$\boldsymbol{M}_1 = \begin{bmatrix} 1-p_1 & p_1 \end{bmatrix} \ ,$$
$$\boldsymbol{M}_2 = \begin{bmatrix} 1-p_2 & p_2 & 0 & 0 \\ 0 & 0 & 1-p_2 & p_2 \end{bmatrix} \ .$$

The other matrices follow analogously. Suppose we want to compute q_1. At the last level, five of the 13 metric vectors in $\mathcal{M}_N$ are members of the set $\mathcal{M}_1^{\mathrm{e}}$, and two of them have nonzero probability $P(\boldsymbol{\mu}_N | p_n = 0)$, namely

$$(1-p_2)p_3(1-p_4)p_5$$
$$\text{and} \quad (1-p_2)p_3p_4p_5 + p_2p_3(1-p_4)p_5 + p_2p_3p_4 \ ,$$

which we sum up to

$$q_1 = p_3p_5 - p_3p_4p_5p_2 + p_2p_3p_4$$

according to (14). Repeating this for all output symbols we obtain (4). □

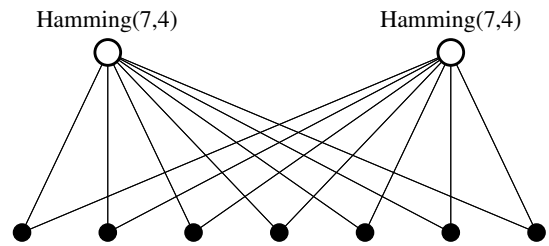

Fig. 3. Ensemble I: Regular protograph of rate $R = 1/7$.

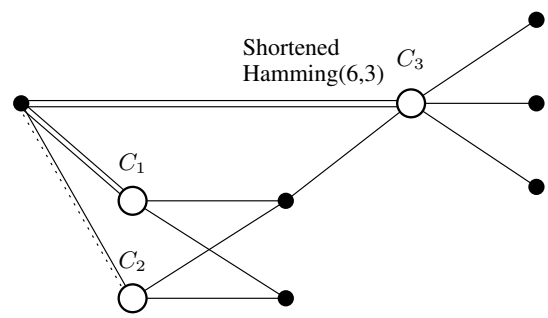

Fig. 4. Ensemble II: Hamming-doped protograph of rate $R = 1/6$.

The reader may also verify that we arrive at the LDPC code density evolution equation (3) by choosing as $\boldsymbol{H}$ an all-one row vector of length K_m.

IV. Application to Some Protograph Examples

As our first example, we consider an $M = 2$, $N = 7$ regular protograph with (7,4) Hamming codes at all constraint nodes, as shown in Fig. 3. All variable nodes have degree two and the rate is equal to $R = 1/7$. We have computed the threshold $\varepsilon^* = 0.7564$ for this code ensemble (Ensemble I) by recursively applying (1) and (2) for different channel parameters ε. The Shannon limit for this code rate is equal to $1 - R = 0.8571$. Due to the regular structure of the graph, an unstructured ensemble with the same degree distributions will have the same threshold. This result can also be obtained by a conventional one-dimensional EXIT chart analysis.

For Ensemble II we choose a "Hamming-doped" protograph from [7], which is depicted in Fig. 4. Starting point for the design of this code was an accumulate-repeat-accumulate (ARA) code. One check node of the protograph of this code was then "doped" by associating a shortened (6,3) Hamming code to it. Depending on whether the dashed edge is excluded from the graph or not, the threshold is equal to $\varepsilon^* = 0.8122$ (Type I) or $\varepsilon^* = 0.8026$ (Type II), respectively ($1 - R = 0.833$). Although it has been observed in [8] that codes in Ensemble II asymptotically do not exhibit a linear growth of stopping set number or minimum distance, very good performance has been observed for practical code lengths [7]. For the unstructured ensemble corresponding to Ensemble II, decoding convergence is prevented by the variable nodes of degree one.

Another example of a protograph with a mixture of SPC and Hamming constraints is given by Ensemble III, shown

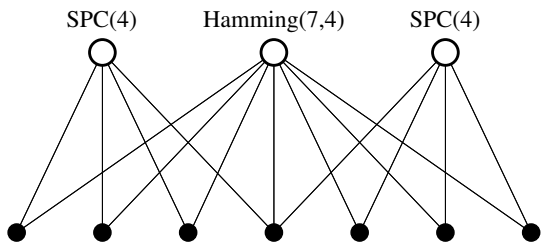

Fig. 5. Ensemble III: Protograph of rate $R = 2/7$.

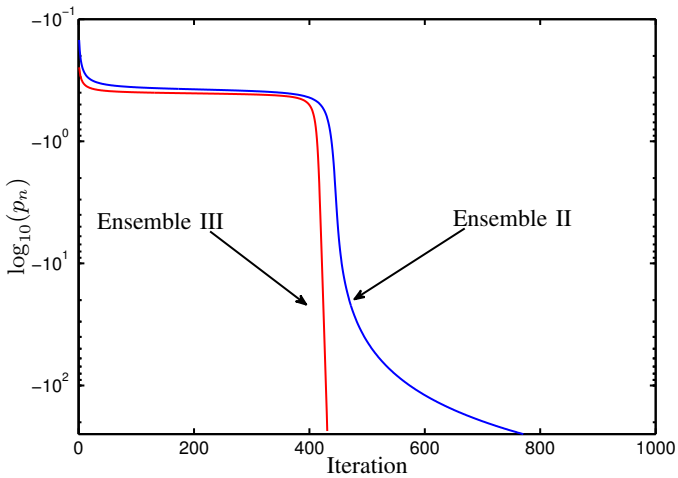

Fig. 6. Convergence behavior: Erasure probabilities p_n as function of the decoding iteration for an ε close to the threshold.

in Fig. 5, with a threshold at $\varepsilon^* = 0.6561$, which can be compared to $1-R = 0.7143$. Interestingly, as shown in Fig. 6, the convergence behavior of Ensemble II (shown for Type II) at low output erasure probabilities looks different from that of Ensemble III. This could be related to the inferior asymptotic distance properties. Actually, Ensemble III is very similar to the codes considered in Example 3 in [8], which asymptotically have linear growth of minimum distance and stopping set number. Interestingly, the unstructured ensemble with same degree distributions as Ensemble III has a better threshold at $\varepsilon^* = 0.6801$. This suggests that there should exist other protographs with better thresholds and we can conclude that in general a careful selection of the particular graph structure is required for best performance.

V. Conclusion

We have proposed a method for calculating analytic expressions for the multi-dimensional input/output transfer functions of APP decoded block codes on the BEC. These can be easily converted to corresponding symbol-wise or averaged EXIT functions for block codes. The computed transfer functions can be used to obtain explicit density evolution equations for protograph GLDPC codes, which can serve as a valuable tool for the design and analysis of efficient code ensembles.

References

[1] 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.

[2] 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.

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

[4] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Global Telecommunications Conference, 2007. GLOBECOM '07.*, Nov. 2007, pp. 3250–3254.

[5] M. Lentmaier, G. 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.

[6] 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.

[7] 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.

[8] S. Abu-Surra, W.E. Ryan, and D. Divsalar, "Ensemble enumerators for protograph-based generalized LDPC codes," *IEEE Global Telecommunications Conference, 2007. GLOBECOM '07.*, pp. 1492–1497, Nov. 2007.

[9] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[10] T. Helleseth, T. Klove, and V.I. Levenshtein, "On the information function of an error-correcting code," *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 549–557, Mar 1997.

[11] E. Paolini, M. Fossorier, and M. Chiani, "Doubly-generalized LDPC codes: Stability bound over the BEC," *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 1027–1046, March 2009.

[12] M. R. Best, M. V. Burnashev, Yannick Y., A. Rabinovich, P. C. Fishburn, A. R. Calderburn, and D. J. Costello, Jr., "On a technique to calculate the exact performance of a convolutional code," *IEEE Transactions on Information Theory*, vol. IT-41, no. 2, pp. 441–447, March 1995.

[13] M. Lentmaier, D.V. Truhachev, and K.Sh. Zigangirov, "Analytic expressions for the bit error probabilities of rate-1/2 memory 2 convolutional encoders," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1303–1311, June 2004.

[14] B.M. Kurkoski, P.H. Siegel, and J.K. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel," *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03.*, vol. 3, pp. 1741–1745 vol.3, Dec. 2003.

[15] T. Johansson and K. Sh. Zigangirov, "A simple one-sweep algorithm for optimal APP symbol decoding of linear block codes," *IEEE Transactions on Information Theory*, vol. IT-44, no. 7, pp. 3124–3129, Nov. 1998.

[16] M. Lentmaier, "Soft iterative decoding of generalized low-density parity-check codes based on MAP decoding of component Hamming codes," Diploma Thesis, 1997, Lund University, Sweden/Univeristy of Ulm, Germany.