From Product Codes to Structured Generalized LDPC Codes

Michael Lentmaier*, Gianluigi Liva†, Enrico Paolini‡, and Gerhard Fettweis*
*Vodafone Chair Mobile Communications Systems, Dresden University of Technology (TU Dresden), 01062 Dresden, Germany
†Institute of Communications and Navigation, German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Wessling, Germany
‡DEIS, WiLAB, University of Bologna, 47023 Cesena (FC), Italy
Emails: {michael.lentmaier, fettweis}@ifn.et.tu-dresden.de, Gianluigi.Liva@dlr.de, e.paolini@unibo.it

***Abstract*—Product codes, due to their relativley large minimum distance, are often seen as a natural solution for applications requiring low error floors. In this paper, we show by means of an ensemble weight enumerator analysis that the minimum distance multiplicities of product codes are much higher than those obtainable by other generalized LDPC (GLDPC) constructions employing the same component codes. We then propose a simple construction of quasi-cyclic GLDPC codes which leads to significantly lower error floors while leaving the decoder architecture of product codes almost untouched.**

***Index Terms*—Iterative decoding, product codes, turbo codes, low-density parity-check codes.**

I. Introduction

Product codes were introduced by Elias in 1954 [1] as a practical coding scheme capable (in an asymptotic setting) to achieve an arbitrarily small error probability on the binary symmetric channel at a code rate bounded away from zero.

Product codes are currently part of the IEEE 802.16 Standard, which foresees 2-dimensional product codes based on various combinations of extended Hamming and single parity-check (SPC) codes[1]. A proposal for the adoption of product codes in the Consultative Committee for Space Data Systems (CCSDS) standard for deep-space mission was presented in [2]. Product codes have been included also in standards for power line communications [3].

Although product codes are usually seen as the "block" counterpart of the original turbo codes [4], they can be seen also from a different perspective. More specifically, product codes can be regarded as generalization of Gallager's low-density parity-check (LDPC) codes [5], where the SPC check nodes are replaced by more powerful constraint (check) nodes (CNs) [6]. Actually, product codes can be regarded as a structured type of generalized LDPC (GLDPC) codes [6]–[8].

The aim of this paper is to present GLDPC codes as an alternative construction of product-like codes and to demonstrate that performance improvements can be already achieved with simple modifications to the product code structure.

[1]Shortened product codes are foreseen by the IEEE 802.16 Standard, meaning that part of the information array $\mathbf{U}$ is 0-padded. The extended Hamming codes allowed as component codes have parameters $(16, 11)$, $(32, 26)$ and $(64, 57)$. The SPC codes allowed as component codes have parameters $(8, 7)$ and $(16, 15)$.

II. Product Codes

A. Code Structure

The structure of a $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$ product code can be conveniently summarized by the encoding procedure. The k information bits are organized in a $k_2 \times k_1$ array $\mathbf{U}$ (with $k_1 k_2 = k$). Each row of $\mathbf{U}$ is then encoded via an (n_1, k_1) binary linear block code $\mathcal{C}_1$. The resulting $k_2 \times n_1$ array is then encoded column-wise through an (n_2, k_2) binary linear block code $\mathcal{C}_2$, leading to an $n_2 \times n_1$ array $\mathbf{C}$ with the structure

$$\mathbf{C} = \left[\begin{array}{c|c} \mathbf{U}_{k_2 \times k_1} & \mathbf{P}^{(1)}_{k_2 \times n_1 - k_1} \\ \hline \mathbf{P}^{(2)}_{n_2 - k_2 \times k_1} & \mathbf{P}^{(12)}_{n_2 - k_2 \times n_1 - k_1} \end{array}\right] . \qquad (1)$$

The length and dimension of the product code are $n = n_1 n_2$ and $k = k_1 k_2$, respectively, while its code rate r is the product of the code rates of $\mathcal{C}_1$ and $\mathcal{C}_2$, i.e., $r = (k_1/n_1)(k_2/n_2) = r_1 r_2$. The two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are usually referred to as *component codes*. It is readily shown that each row of $\mathbf{C}$ is a codeword of $\mathcal{C}_1$. Similarly, each column is a codeword of $\mathcal{C}_2$. Typical component codes are short algebraic codes such as SPC, Hamming, Bose-Chaudhuri-Hochquenghem (BCH) (and their shortened/extended versions) codes [9]–[11]. The above description refers to the case of a 2-dimensional product code. Product codes may be built on arrays of higher dimensions. However, we will stick to the 2-dimensional case in the following.

Product codes may be decoded iteratively by means of soft-input soft-output (SISO) decoding of their component codes [1], [6], [11], [12], in which case they are often referred to as *block turbo codes (BTCs)* or *turbo product codes (TPCs)*.

B. Distance Spectrum

Denoting by d_1 and d_2 the minimum distances of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, the minimum distance of the product code is given by $d_{\min} = d_1 d_2$. In fact, we have $d_{\min} \geq d_1 d_2$ since any non-zero codeword $\mathbf{C}$ contains at least one row of weight $w_r \geq d_1$, and each '1' in this row implies in turn a column of weight $w_c \geq d_2$. Moreover, a weight-$d_1 d_2$ codeword can be obtained as follows:

- select a weight-d_1 codeword $\mathbf{c}^{(1)} \in \mathcal{C}_1$ and a weight-d_2 codeword $\mathbf{c}^{(2)} \in \mathcal{C}_2$;
- build a $n_2 \times n_1$ array $\tilde{\mathbf{C}} = (\mathbf{c}^{(2)})^T \mathbf{c}^{(1)}$.

TABLE I
MINIMUM DISTANCES AND MULTIPLICITIES OF SOME SELECTED PRODUCT CODES (EH = EXTENDED HAMMING CODE, SPC = SINGLE PARITY CHECK CODE).

(n,k)	$\mathcal{C}_1$	$\mathcal{C}_2$	$d_{\min}$	$A_{d_{\min}}$
$(256,121)$	eH $(16,11)$	eH $(16,11)$	16	19600
$(256,165)$	eH $(16,11)$	SPC $(16,15)$	8	16800
$(256,225)$	SPC $(16,15)$	SPC $(16,15)$	4	14400
$(1024,676)$	eH $(32,26)$	eH $(32,26)$	16	1537600
$(1024,806)$	eH $(32,26)$	SPC $(32,31)$	8	615040
$(1024,961)$	SPC $(32,31)$	SPC $(32,31)$	4	246016
$(4096,3249)$	eH $(64,57)$	eH $(64,57)$	16	108493056
$(4096,3591)$	eH $(64,57)$	SPC $(64,63)$	8	20998656
$(4096,3969)$	SPC $(64,63))$	SPC $(64,63)$	4	4064256

It can be verified that $\tilde{\mathbf{C}}$ satisfies the constraints of all rows and columns. While the minimum distance of a product code is known (provided d_1 and d_2 are), the characterization of its complete distance spectrum still represents an open problem. Let us denote by $A^{(1)}(s) = 1 + \sum_{i=d_1}^{n_1} A_i^{(1)} s^i$ and $A^{(2)}(s) = 1 + \sum_{i=d_2}^{n_2} A_i^{(2)} s^i$ the weight enumerator functions (WEFs) of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and by $A(s) = 1 + \sum_{i=d_{\min}}^{n} A_i s^i$ the WEF of the corresponding product code. Expressing $A(s)$ as function of $A^{(1)}(s)$ and $A^{(2)}(s)$ represents a very appealing result. Approaches to solve this problem, which comes out to be extremely complex, have been proposed in [13]–[15], where an exact expression of the WEF for the low-weight codewords and an approximate expression of the WEF for the higher-weight codewords are developed. It was shown in [14] that the multiplicity of codewords with minimum (non-zero) Hamming weight of a product code is equal to the product of the minimum distance multiplicities of its component codes, i.e., $A_{d_{\min}} = A_{d_1}^{(1)} A_{d_2}^{(2)}$. We shall see later that the knowledge of $A_{d_{\min}}$ provides much of the information needed to characterize the code performance in the low error rate regimes.

In some cases, the multiplicities of the coefficients of $A^{(1)}(s)$ and $A^{(2)}(s)$ can be conveniently obtained through the MacWilliams identity [16]. The minimum distance multiplicity (i.e., the multiplicity of codeword with Hamming weight equal to code minimum distance) of a Hamming code can be easily calculated as $A_3 = \binom{n}{2}/3$. A thorough analysis of the parameters for product codes based on extended and shortened Hamming codes is provided in [10], [17].

In Table I, some product codes based on SPC and extended Hamming codes are listed, together with their main parameters.[2] Extended Hamming codes are commonly used in place of Hamming codes to construct product codes, due to their larger minimum distance (obtained at the price of a slight rate loss).

C. Error Floor Analysis

The performance over the additive white Gaussian noise (AWGN) channel of a $(1024,676)$ product code where both $\mathcal{C}_1$ and $\mathcal{C}_2$ are $(32,26)$ extended Hamming codes, under soft iterative decoding with Bahl-Cocke-Jelinek-Raviv (BCJR) decoding at the component codes [18], is depicted in Fig. 4. The performance is in terms of codeword error rate (CER) vs. E_b/N_0 (where E_b is the energy per information bit and N_0 the one-sided noise power spectral density) and assumes antipodal signaling. The maximum number of decoding iterations has been set to 20. A prediction of the product code error floor is also shown in Fig. 4. It is based on the union bound (UB) on the block error probability

$$P_B \leq \sum_{i=d_{\min}}^{n} A_i \, Q\left(\sqrt{2\, i\, r\, \frac{E_b}{N_0}}\right) \tag{2}$$

where $Q(x) = \left(\sqrt{2\pi}\right)^{-1} \int_x^\infty e^{-t^2/2}\, \mathrm{d}t$. For large signal-to-noise ratios (SNRs) P_B may be approximated by the dominating term that is associated with $i = d_{\min}$ [19]. For the $(1024,676)$ product code this truncated UB is depicted in Fig. 4.[3] The CER performance for the product code under plain iterative decoding does not approach the truncated UB, but tends to remain almost one order of magnitude larger. As noted in [10], [11], a sensible performance improvement can be obtained by scaling the extrinsic information at the output of each component decoder. The performance curve using this weighted extrinsic information (w.e.i.) approach (with scaling factor set to 0.5) is depicted in Fig. 4. In this case, at high SNR the CER tightly approaches the error floor prediction. Indeed, the floor appears at a rather high error rate, i.e. at $\mathsf{CER} \simeq 10^{-4}$. In fact, while for higher error rates the performance is within 0.6 dB from the random coding bound (RCB) [5], a lower error rates we observe a remarkable coding gain loss. According to the error floor prediction, at $\mathsf{CER} = 10^{-7}$ the loss w.r.t. the RCB is nearly 1.5 dB. The high error floor is due to the huge value of $A_{d_{\min}}$ as from Table I, playing a fundamental role in the truncated UB [17]. We observe that this issue affects product codes in general. On one side, they exhibit relatively large minimum distances thanks to the $d_{\min} = d_1 d_2$ relationship. On the other hand, the minimum distance multiplicities are usually very high.

Despite of this problem, product codes have been often regarded as a natural solution for applications requiring low error floors [20], [21]. Product codes have been frequently considered for many applications also thanks to the possibility of implementing efficient decoder architectures see for instance [22].

III. PRODUCT CODES AS INSTANCES OF GLDPC CODES

A. Tanner Graph Representation of GLDPC Codes

An LDPC code is conveniently represented in a graphical fashion by means of a bipartite graph (or *Tanner graph* [6]) with two disjoint sets of nodes, namely, the variable nodes (VNs) and the CNs, such that each edge is only allowed to connect a VN with a CN. In the Tanner graph of an LDPC code, the VNs have a one-to-one correspondence with the encoded bits and the CNs with the parity-check equations.

[2]Note that for all product codes in Table I, $\mathcal{C}_1$ and $\mathcal{C}_2$ coincide. This is however not necessary in general.

[3]We remark that the truncated UB does not represent an upper bound to the block error probability but only an approximation in the high SNR regimes. Moreover, the bound of (2) holds under maximum likelihood (ML) decoding, while numerical results are provided for iterative decoding only.

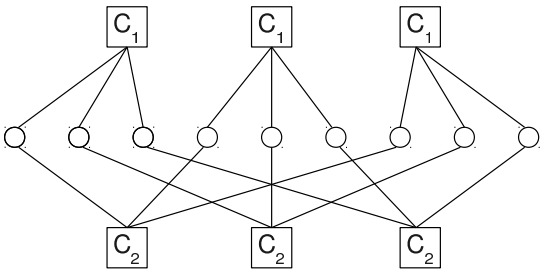

Fig. 1. Tanner graph of a length-9 product code where both component codes $\mathcal{C}_1$ and $\mathcal{C}_2$ have length 3. Each VN is checked by a $\mathcal{C}_1$ code and by a $\mathcal{C}_2$ code. Note that the dimension of the length-9 product code depends on the dimensions of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Therefore, the parity-check matrix of an LDPC code coincides with the adjacency matrix $\mathbf{\Gamma}$ of its Tanner graph. A Tanner graph is called sparse if the density of its adjacency matrix, defined as the fraction of non-zero elements in $\mathbf{\Gamma}$, is smaller than one half. As a consequence, the Tanner graph of an LDPC code is sparse. In the Tanner graph, the degree of a node is defined as the number of edges incident on it, and the girth g of the graph is defined as the length of its shortest cycle. Note that a degree-$\tilde{n}$ CN of an LDPC code may be interpreted as a length-$\tilde{n}$ SPC code, as it checks the parity of the $\tilde{n}$ VNs connected to it.

Even prior to the discovery of turbo codes, GLDPC codes were introduced by Tanner in 1981 [6]. Analogously to an LDPC code, also a GLDPC code is represented by a sparse Tanner graph with a relatively small number of edges. A GLDPC code generalizes the concept of an LDPC code in that a degree-$\tilde{n}$ CN may in principle be any $(\tilde{n}, \tilde{k})$ linear block code, where $\tilde{n}$ is the code length and $\tilde{k}$ the code dimension. Such an $(\tilde{n}, \tilde{k})$ code is usually referred to as *component code* and the corresponding CN as an $(\tilde{n}, \tilde{k})$ CN. An $(\tilde{n}, \tilde{k})$ CN has $\tilde{n}$ connections towards the VNs and accounts for $\tilde{n} - \tilde{k} \geq 1$ linearly independent parity-check equations. A binary sequence is a codeword for the GLDPC code if and only if each CN recognizes one of its local codewords. In [6] *regular* GLDPC codes (also known as *Tanner codes*) were investigated, these being GLDPC codes where the VNs have all the same degree and the CNs are all linear block codes of the same type.

Product codes introduced in Section II admit a very simple representation in terms of Tanner graph. According to this representation, product codes may be regarded a special subclass of GLDPC codes. The Tanner graph of a product code comprises $2n_1n_2$ edges, a set of $n = n_1n_2$ VNs and a set $C = C_1 \cup C_2$ of $n_2 + n_1$ CNs such that $|C_1| = n_2$ and $|C_2| = n_1$. Each of the n_2 CNs belonging to C_1 has degree n_1 and imposes the $n_1 - k_1$ constraints of the component code $\mathcal{C}_1$ to its neighboring VNs, whereas each of the n_1 CNs belonging to C_2 has degree n_2 and imposes the $n_2 - k_2$ constraints of the component code $\mathcal{C}_2$ to its neighboring VNs. Each VN is connected to a constraint node in C_1 and to one in C_2 (in fact, each bit in the array $\mathbf{C}$ of (1) has to fulfill both a row and a column constraint), and therefore all VNs in the graph have degree 2. Note that all the n_1 VNs corresponding to the bits in a generic row of $\mathbf{C}$ are connected to a unique CN in C_1 and that each of these VNs is connected to a specific CN in C_2. Similarly, all the n_2 VNs corresponding to the bits in a generic column of $\mathbf{C}$ are connected to a unique CN in C_2 and each of these VNs is connected to a specific CN in C_1. As a result, it is readily shown that the Tanner graph of any product code possesses a girth $g = 8$. For example, in Fig. 1 the Tanner graph of a simple product code of length $n = 9$, where both $\mathcal{C}_1$ and $\mathcal{C}_2$ are length-3 SPC codes, is depicted. The graph has nine VNs and six CNs. For ease of representation, the three CNs belonging to C_1 and the three CNs belonging to C_2 are drawn above and below the VNs, respectively. The adjacency matrix of the product code in Fig. 1 is equal to

$$\mathbf{\Gamma} = \begin{bmatrix} 1&1&1&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1 \\ 1&0&0&1&0&0&1&0&0 \\ 0&1&0&0&1&0&0&1&0 \\ 0&0&1&0&0&1&0&0&1 \end{bmatrix} . \tag{3}$$

It is easy to check that the graph has girth $g = 8$.

We point out that there exist other classes of codes that may be seen as simple special instances of GLDPC codes. Among them, we mention *expander codes* constructed on sparse bipartite graphs investigated, for instance, in [23], [24]. Here, each node in the bipartite graph is associated with a binary linear code, and each edge with an encoded bit, such that a binary word is a valid codeword for the expander code if and only if each node in the graph recognizes a valid local codeword. Using a procedure identical to that described in [25, Sec. IV-A], such an expander code may be always represented as a GLDPC code where all VNs have a degree 2.

It is worthwhile observing that the concept of GLDPC code may be generalized even further by allowing the VNs as well as the CNs to be of any generic linear block code types. The obtained code structure is known to be a doubly-generalized LDPC (D-GLDPC) code, and allows a higher design flexibility, especially in terms of code rate [26]–[28]. A degree-$\tilde{n}$ VN may in principle be any $(\tilde{n}, \tilde{k})$ linear block code, where $\tilde{n}$ is the code length and $\tilde{k}$ the code dimension. Such a VN is associated with $\tilde{k}$ encoded bits of the D-GLDPC code. It interprets these bits as its local information bits and interfaces to the CN set through its $\tilde{n}$ local code bits. Local encoding at the VNs may be either systematic or non-systematic. Note that a D-GLDPC code whose VNs perform a local systematic encoding may be interpreted as a punctured expander code (or as a punctured GLDPC code), where the punctured bits are those associated with the local parity bits of each VN. This interpretation is however limited to the case where all VNs of the D-GLDPC code are in systematic form and does not hold for the iterative decoders.

B. Structured and Unstructured Ensembles of GLDPC Codes

For product codes, it follows from the Tanner graph structure that the density of the adjacency matrix $\mathbf{\Gamma}$ is equal to $2/(n_1 + n_2)$ and thus decreases with the lengths of the

component codes, which define the degrees of the CNs. More generally, longer GLDPC codes of lower density can be obtained not only by increasing the length of the component codes but by increasing the number of VNs and CNs while keeping the component codes and variable node degrees fixed. We distinguish between ensembles of *structured* and *unstructured* codes. Consider the construction of a GLDPC code of length n, where the CN set is composed of a mixture of two component code types $\mathcal{C}_1$ and $\mathcal{C}_2$ of equal length $n_1 = n_2 = \tilde{n}$ and each symbol is protected by two component codes. The regular Tanner graph of such a code has n VNs of degree two and $|C| = n/n_1 + n/n_2 = 2n/\tilde{n}$ CNs of degree $\tilde{n}$. The density of $\mathbf{\Gamma}$ is equal to $\tilde{n}/n$ and decreases with the number of VNs. The corresponding ensemble of unstructured GLDPC codes is defined by the set of Tanner graphs that can be obtained by all $(2n)!$ possible permutations of edges. The design rate of the ensemble is equal to $r = r_1 + r_2 - 1$.

Alternatively, a small Tanner graph called *protograph* [29] can be used as a template for the construction of longer GLDPC codes [30]. The adjacency matrix of a protograph is called its *base matrix* $\mathbf{B}$ and the adjacency matrix $\mathbf{\Gamma}$ of a GLDPC code can be derived from this base matrix by replacing each 1 in $\mathbf{B}$ by a permutation matrix and each 0 by an all-zero matrix. For example, the base matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

represents a compact regular protograph for the case $\tilde{n} = 15$. More generally, starting from a size $2 \times \tilde{n}$ all-one base matrix $\mathbf{B}$, an ensemble of length n regular protograph-based GLDPC (PG-GLDPC) codes can be defined by the set of Tanner graphs resulting from the $2\tilde{n}(n/\tilde{n})!$ possible choices of permutation matrices. By this construction each node in the protograph is replicated $n/\tilde{n}$ times and the edges are permuted among these replica in such a way that the structure of the original graph is preserved. The resulting PG-GLDPC codes form therefore an ensemble of structured codes of design rate $r = r_1 + r_2 - 1$.

Also the structural properties of product codes can be captured by means of protographs. Using the Tanner graph of a product code as protograph, we can obtain ensembles of PG-GLDPC codes with flexible blocklengths n. On the other hand, after reordering of columns, the adjacency matrix $\mathbf{\Gamma}$ of a product code can be considered as a particular instance in a regular PG-GLDPC code ensemble as introduced above. For example, the reader may verify that a reordered version of the matrix $\mathbf{\Gamma}$ in (3) can be derived from a 2×3 all-one base matrix $\mathbf{B}$ by replacing each one by a permutation matrix of size three. Examples of quasi-cyclic regular PG-GLDPC codes in comparison with product codes are presented in Section V.

The ensemble definitions can be extended to irregular graphs in which the node degrees are not fixed. In the structured case this is achieved by means of a base matrix $\mathbf{B}$ with columns and rows of different weights. As a particular example, the shortened product codes, achieved by a zero-padded information array, can be described by such an irregular base matrix, which results from a removal of the columns associated with the padded symbols. In the case of unstructured irregular ensembles, the degrees of VNs and CNs are then considered as random variables and characterized by their degree distributions, which define the fractions of edges incident to VNs and CNs of a certain degree.

IV. Performance of GLDPC Code Ensembles

A. Weight Distribution of GLDPC Codes

As for product codes, the derivation of the WEF for a specific GLDPC code represents a very hard task. The problem may be somehow circumvented by computing the average WEF for the finite-length GLDPC code ensemble. As an example, let us consider the case of a regular GLDPC ensemble with n degree-2 VNs and m CNs, all of the same type. Let us denote the WEF of the generic CN by $A^{(1)}(s)$. Moreover, let us assume that the CN set is divided into two disjoint subsets with $m/2$ CNs each, such that every VN is checked by one CN in the first subset and by one CN in the second subset. Taking an approach similar to that described in [31] within the context of LDPC codes, it is possible to show that the average WEF of the GLDPC ensemble, denoted by $\bar{A}(s) = \sum_l \bar{A}_l s^l$, is such that

$$\bar{A}_l = \frac{\left(\text{coeff}\left((A^{(1)}(s))^{m/2}, s^l\right)\right)^2}{\binom{n}{l}} \tag{5}$$

where $\text{coeff}(g(x), x^l)$ denotes the coefficient of x^l in the polynomial $g(x)$, and where $\bar{A}_l = \mathsf{E}A_l$ denotes the expected number of codewords of Hamming weight l for a code randomly drawn from the ensemble with uniform probability. The average WEF can be then used together with the UB (2) to obtain an upper bound on the expected block error probability over the ensemble. For instance, in Fig. 2 the UB for such a GLDPC *unexpurgated ensemble* whose Tanner graph has 1024 VNs and 64 CNs based on the $(32, 26)$ extended Hamming code is compared with the truncated UB for the $(1024, 676)$ product introduced in Section II. Note that this product code belongs to the considered GLDPC ensemble. At high SNRs, the average UB of the GLDPC code ensemble is affected by low-weight codewords, for which the average multiplicity $\bar{A}_l$ is small, but not yet zero. In other words, the average performance of the GLDPC ensemble at high SNRs is dominated by a subset of codes with bad distance properties.

Instead of considering the expected WEF over the whole GLDPC code ensemble, we may restrict the analysis to an *expurgated ensemble* using again a technique similar to that described in [31]. Given the probability $p_l = \bar{A}_l / \binom{n}{l}$ that a weight-l binary sequence of length n is a codeword, an upper bound to the cumulative probability function for the minimum distance of a code randomly picked in the ensemble is given by

$$\Pr\{d_{\min} \leq D\} \leq \sum_{l=1}^{D} \binom{n}{l} p_l \triangleq F(D)\,.$$

Let us define δ as the maximum positive integer such that $F(\delta) \leq 1/2$. Then, for a code randomly picked in the ensemble we have $\alpha \triangleq \Pr\{d_{\min} \leq \delta\} \leq 1/2$, meaning

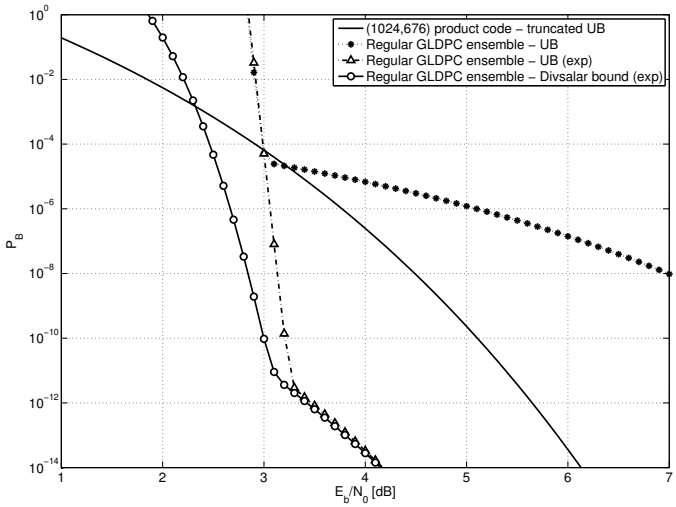

Fig. 2. UBs and Divsalar bound [32] on the expected block error probability of a GLDPC code with 1024 degree-2 VNs and 64 CNs based on the $(32, 26)$ extended Hamming code, randomly selected in the unexpurgated and expurgated ensembles, vs. the truncated UB for the $(1024, 676)$ product code based on the same component codes.

that a fraction of at least $1 - \alpha > 1/2$ of the codes in the ensemble has minimum distance larger than δ. Let us denote by $\bar{A}'(z)$ the average WEF over the subset of *bad* codes in the ensemble (i.e., the expected WEF for a code randomly picked in the subset of codes for which $d_{\min} \leq \delta$), and by $\bar{A}''(z)$ the WEF for the *good* codes (i.e., the expected WEF for a code randomly drawn in the subset of codes for which $d_{\min} > \delta$). The coefficient $\bar{A}_l$ may be expressed as

$$\bar{A}_l = \alpha \bar{A}'_l + (1 - \alpha)\bar{A}''_l$$

so that, for $l > \delta$ we can write

$$\bar{A}''_l = \frac{\bar{A}_l}{1 - \alpha} - \frac{\alpha \bar{A}'_l}{1 - \alpha}$$

which is maximized by choosing $\bar{A}'_l = 0$ and $\alpha = 1/2$ (recall that $\alpha \leq 1/2$). It follows that we can upper bound the expected multiplicity of codewords of weight $l > \delta$ for a code randomly chosen among the the good codes as $\bar{A}''_l \leq 2\bar{A}_l$. For our example GLDPC ensemble, the first non-zero coefficient of $\bar{A}''(z)$ is the coefficient of z^{16}, meaning that the minimum distance of any GLDPC code among the expurgated ensemble is lower bounded by 16.

The average UB for the expurgated ensemble is provided in Fig. 2, together with an even tighter upper bound on the block error probability derived in [32]. Interestingly, for the good codes forming the expurgated ensemble, the multiplicity of codewords of weight 16 is remarkably lower than that of the product code. This is the reason for the huge gap in the error floor performance between the average UB for expurgated GLDPC ensemble and the truncated UB for the product code. Note also that any such GLDPC code whose Tanner graph has girth $g \geq 8$ must belong to the expurgated ensemble, as its minimum distance is necessarily lower bounded by 16. In fact, according to the simple tree bound presented in [6, Theorem 2], the minimum distance $d_{\min}$ of any GLDPC code $\mathcal{C}$ with VN degree 2 and $g \geq 8$ is lower bounded as $d_{\min} \geq d + d(d-1) = d^2$, where d is the minimum distance of each component code. We will show later how to build explicitly a Tanner graph fulfilling this condition.

The design of finite-length GLDPC codes (LDPC codes as a special case) has often benefited from asymptotics. Asymptotic tools allow to predict, on a statistical basis, the performance offered by an LDPC code randomly drawn from an ensemble with given characteristics, in the limit where the codeword length tends to infinity. Examples of asymptotic tools of common use are the *decoding threshold* [33] (related to the waterfall performance, see also Sec. IV-B) and the *critical exponent codeword weight ratio* [34] (related to the error floor performance), briefly reviewed next.

Let us consider a sequence of unstructured or structured GLDPC code ensembles. All the ensembles in the sequence share common features: for example, VN degree profile, CN types and distribution in the unstructured case, protograph in the structured case. Each ensemble in the sequence is associated with a codeword length n. The expected number of codewords of linear weight ωn of a length-n GLDPC code randomly picked in the corresponding ensemble of the sequence may be expressed as $\mathsf{E}A_{\omega n} \approx e^{G(\omega)n}$ for large n. The function $G(\omega)$ is known as the *growth rate of the weight distribution* or as the *spectral shape* of the ensemble sequence [31]. The critical exponent codeword weight ratio, here denoted by ω^*, is defined as

$$\omega^* = \inf\{\omega > 0 : G(\omega) \geq 0\}.$$

If $\omega^* = 0$ then a length-n GLDPC code randomly picked in the corresponding ensemble of the sequence exhibits an exponentially large number of small linear-sized codewords. Therefore, it has bad minimum distance properties with high probability, even for very large n. On the other hand, if $\omega^* > 0$ then the expected number of linear-sized codewords of normalized weight $0 < \omega < \omega^*$ for a length-n code picked in the corresponding ensemble tends to zero exponentially as n tends to infinity. It turns out that GLDPC codes with good minimum distance properties should be searched for in (structured or unstructured) ensembles characterized by $\omega^* > 0$, that is by *good spectral shape behavior*.

A complete solution for the spectral shape of unstructured GLDPC ensembles has been developed in [35], [36], while a complete analysis of $G(\omega)$ for small values of ω in unstructured ensembles is available in [37]. For structured GLDPC ensembles based on protographs and for GLDPC ensembles where each VN is checked by a CN of one type and by a CN of another type we refer to [38] and [25], respectively.

B. Iterative Decoding of GLDPC Codes

Decoding of GLDPC codes is based on the belief-propagation principle, and is performed through iterative message passing over the Tanner graph. Each decoding iteration consists of an exchange of messages between the VNs and the CNs. In a first half-iteration, extrinsic log-likelihood ratios (LLRs) are sent from the VNs to the CNs along the edges of

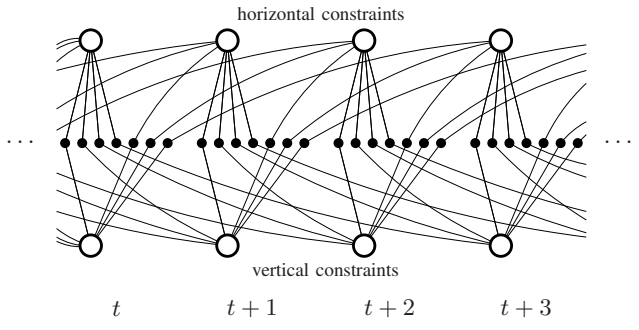

Fig. 3. Tanner graph of a BBC with (7,4) Hamming component codes. The nodes are grouped according to the time instant at which the code symbols are generated.

the graph. Each CN interprets the received soft messages as an *a priori* information and generates the corresponding extrinsic LLRs: to this purpose, the same SISO algorithms employed at the component codes of a product code (such as BCJR [18] or other trellis-based decoding) are used. In a second half-iteration, extrinsic LLRs are sent back from the CNs to the VNs. Each VN interprets the received soft messages as an *a priori* information and generates the corresponding extrinsic LLRs. At the end of each iteration, the *a posteriori* probability information is calculated for each encoded bit, and a hard decision is taken. If the obtained binary sequence is a valid codeword, then a success is declared. Otherwise either a new iteration is started or, if a maximum number of iterations has been reached, a decoding failure is declared.

The belief propagation threshold is a measure for the asymptotically achievable performance of iterative decoding. It is the worst case channel parameter for which the decoding error probability converges to zero with the number of iterations for a specific class of codes. For the computation of such thresholds, the probability density functions (PDFs) of the messages exchanged in the decoder can be tracked as function of the iterations (density evolution [33]).

In general it is difficult to express the input/output transfer functions of the component decoders in an analytical way, so that Monte Carlo methods are typically applied for empirical threshold evaluation. In case of unstructured ensembles, the message PDFs along different edges can be averaged over the ensemble and density evolution is reduced to a single-parameter problem. Furthermore, Gaussian approximations of the PDFs are frequently used in practice to simplify the calculations. In this case, a graphical representation of the convergence behavior by means of Extrinsic Information Transfer (EXIT) charts [39] can be a convenient tool not only for threshold evaluation but also for component code matching. As special role plays the binary erasure channel (BEC), for which an exact density evolution analysis can be performed analytically and is equivalent to an EXIT chart analysis [27]. For structured ensembles, however, multi-dimensional transfer functions are required for the characterization of the component decoders [40] and a graphical visualization becomes difficult. For other channels the threshold computation for structured ensembles is therefore still an open problem.

Consider a GLDPC code ensemble based on $(15, 11)$ Hamming component codes as defined by the base matrix (4). Its BEC threshold is equal to $\varepsilon^* = 0.4678$, which is equivalent to the threshold of the corresponding unstructured ensemble because of the regular structure of both the protograph and the component Hamming codes. For the AWGN channel the density evolution threshold can be evaluated by Monte Carlo methods as $(E_b/N_0)^* = 0.86$ dB [41], while an EXIT chart analysis results in $(E_b/N_0)^* = 0.75$ dB or $(E_b/N_0)^* = 1.04$ dB, depending whether the mean or the SNR of the PDFs is fitted at the output of the component decoder [30].

C. GLDPC Convolutional Codes

An example of specially structured ensembles of GLDPC codes are braided block codes (BBCs) [41], [42], which can be defined by means of an infinite, diagonally shaped array. Each code symbol in this array is protected by one horizontal and one vertical component code. BBCs may be interpreted either as a convolutional code version of product codes or, if sparsity is introduced into their structure, as GLDPC convolutional codes. Figure 3 shows the Tanner graph of a BBC with (7,4) Hamming component codes.

Analogously to block codes, an ensemble of GLDPC convolutional codes can be constructed from a protograph. Such PG-GLDPC convolutional codes can be described by means of a *convolutional protograph* with base matrix

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

where m_{cc} denotes the memory of the convolutional codes and the *component base matrices* $\mathbf{B}_i$, $i = 0, \ldots, m_{\mathrm{cc}}$, describe the edges from the VNs at time t to the CNs at time $t + i$. Ensembles of protograph based BBCs can be derived by using the Tanner graph of a BBC as a protograph [43]. Using the Tanner graph in Fig. 3 as an example, the component base matrices can be identified as

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

A density evolution analysis of such PG-GLDPC convolutional codes shows a dramatic threshold improvement compared to the corresponding block code counterparts [43]. In fact, for the BEC it can be shown that the convolutional code thresholds under belief propagation decoding are equal to the block code thresholds under optimal *maximum a posteriori* (MAP) decoding. For the ensemble based on $(15, 11)$ Hamming component codes, the threshold is improved from $\varepsilon^* = 0.4678$ to $\varepsilon^* = 0.5277$, while the Shannon limit is at $\varepsilon^* = 0.5333$.

V. Quasi-Cyclic GLDPC Codes as Alternative to Product Codes

A way of constructing structured quasi-cyclic (QC) GLDPC codes was presented in [30]. The code construction is based on the expansion of a GLDPC protograph by means of circulant permutation matrices. Although the approach of [30] is general and permits do construct both regular and irregular types of GLDPC codes, we present next a simple deterministic design for regular QC GLDPC codes. The construction aims at producing a QC GLDPC with parameters as close as possible to those of a target product code. The proposed approach requires only that the component codes of the target product code share the same coded block size, i.e. that $n_1 = n_2 = \tilde{n}$. Thus, we will indicate as $\mathcal{C}_1$ and $\mathcal{C}_2$ the component codes of the target product code with parameters $(\tilde{n}, k_1)$ and $(\tilde{n}, k_2)$ respectively. The construction starts by building the adjacency matrix of the QC GLDPC with the following form

$$\mathbf{\Gamma} = \begin{bmatrix} \beta^0 & \beta^0 & \beta^0 & \dots & \beta^0 & \beta^0 \\ \beta^0 & \beta^1 & \beta^2 & \dots & \beta^{\tilde{n}-2} & \beta^{\tilde{n}-1} \end{bmatrix} \tag{6}$$

being β a $\tilde{n} \times \tilde{n}$ circulant permutation matrix obtained by the right rotation (by 1 position) of the $\mathbf{I}_{\tilde{n}\times\tilde{n}}$ identity matrix, with $\beta^0 = \mathbf{I}_{\tilde{n}\times\tilde{n}}$.[4] The first "block" row (i.e., the first row of circulants) of $\mathbf{\Gamma}$ is associated to the $\tilde{n}$ CNs based on $\mathcal{C}_1$, the second "block" row to the $\tilde{n}$ CNs based on $\mathcal{C}_2$.

Interestingly, the above-proposed construction provides the same girth of the product code, $g = 8$, and therefore guarantees that the minimum distance of the GLDPC code is lower-bounded by that of the target product code. To see this, we first note that the adjacency matrix (6) is free from cycles of length 4. This is due to the fact that there are no identical columns in Γ. Moreover, regular block-circulant matrices with column weight 2 admit only cycles with lengths multiple of 4 (see Cor.2.1 in [44]). Hence, we obtain a first lower bound on g as $g \geq 8$. The bound is actually met with equality. To show this, it is sufficient to combine the first two "block" columns (i.e., the first column of circulants) of Γ and to compare the result with a combination of the third and the fourth "block" column of Γ. The combination of the first two block columns brings to $\left[\mathbf{0} \;\; (\beta^0 + \beta^1)\right]^T$, while the combination of the third and the fourth block column is given by $\left[\mathbf{0} \;\; (\beta^2 + \beta^3)\right]^T$. It is easy to check that there are $\tilde{n}$ columns in $\left[\mathbf{0} \;\; (\beta^0 + \beta^1)\right]^T$ which have an identical column in $\left[\mathbf{0} \;\; (\beta^2 + \beta^3)\right]^T$.[5] This means that there are sets of 4 columns in Γ that are linearly dependent, and hence that there are cycles of length 8.

The QC GLDPC code parameters are given by $n = \tilde{n}^2$, $k \geq n - (\tilde{n}(\tilde{n} - k_1 + \tilde{n} - k_2))$. Hence, the final code rate may be slightly less than that of the target product code. As an example, if we assume that the two component codes are both a $(32, 26)$ extended Hamming code, we would obtain a product code with parameters $(1024, 676)$ (the code rate is 0.66), while the GLDPC would have $n = 1024$ and $k \geq 640$ (hence, the code rate is ≥ 0.625). In both cases we have $d_{min} \geq 16$. For the unstructured regular GLDPC ensemble with VN degree 2 and CNs based on the $(32, 26)$ extended Hamming code, the EXIT chart threshold can be evaluated as $(E_b/N_0)^* = 1.41$ dB, while the Shannon limit for the binary-input AWGN channel is at $E_b/N_0 = 0.82$ dB.

[4] $\{\beta^0, \beta^1, \dots, \beta^{\tilde{n}-1}\}$ forms a cyclic multiplicative group of order $\tilde{n}$.

[5] $(\beta^2+\beta^3)$ is a cyclic rotation of $(\beta^0+\beta^1)$, i.e. $(\beta^2+\beta^3) = \beta^2(\beta^0+\beta^1)$.

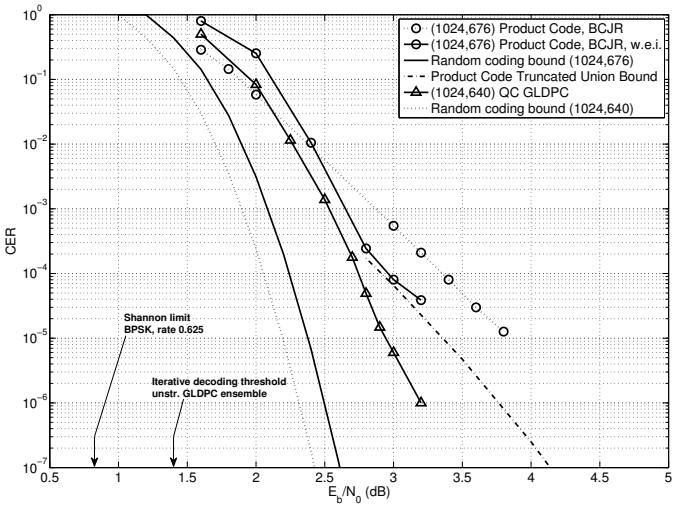

Fig. 4. Performance of the $(1024, 640)$ QC GLDPC code, compared to that of the $(1024, 676)$ product code based on the $(32, 26)$ extended Hamming code. 20 iterations.

The $(1024, 640)$ QC GLDPC code performance are depicted in Fig. 4 in terms of CER vs. E_b/N_0, and are compared to those of the $(1024, 676)$ product code based on the same component codes. Since the QC GLDPC possesses a slightly lower code rate (0.625 vs 0.66), the RCBs for both $(1024, 640)$ and $(1024, 676)$ codes are provided. Hence, when comparing the codes one shall take into account a penalty of ~ 0.1 dB for the GLDPC code due to its lower rate. Accounting for that, the performance of the two codes in the waterfall region are nearly the same (we consider as reference curve for the product code the one obtained using the w.e.i. approach). However, as expected, the GLDPC code exhibits a coding gain at lower error rates w.r.t. the product code. At the last simulation point the GLDPC code achieves a $\mathsf{CER} = 10^{-6}$ without signs of floor at $E_b/N_0 = 3.2$ dB, while the product code would require (according to the error floor estimation) nearly 3.75 dB for the same error rate. The decoding complexity of the GLDPC code is equal to that of the product code and its QC structure allows linear complexity encoding.

VI. Conclusions

It can be observed that the special structure of product codes leads to large minimum distance multiplicities. In this paper, we demonstrated that these large multiplicities have a significant negative influence on the error floors of product codes. We presented an overview on the concept of GLDPC codes and pointed out additional degrees of freedom that can be used in the construction of product-like codes. As a practical example, employing the same component codes we proposed a simple QC GLDPC code construction as alternative

to product codes. This construction shows that with simple modifications to the product code structure it is possible to design GLDPC codes with the same minimum distance but much lower minimum distance multiplicity, resulting in lower error floors while leaving the decoder architecture almost untouched.

ACKNOWLEDGMENTS

This work was supported in part by the EC under Seventh FP grant agreement ICT OPTIMIX n. INFSO-ICT-214625. E. Paolioni would also like to thank Prof. Marco Chiani for useful discussion.

REFERENCES

[1] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. PGIT-4, pp. 29–37, Sept. 1954.

[2] P. Adde, R. Pyndiah, G. Moury, and G. Lesthievent, *Recent Simplifications and Improvements in Block Turbo Codes*, ENST Bretagne/CNES, Oct. 2000, CCSDS Fall Meeting.

[3] M. K. Lee, R. E. Newman, H. A. Latchman, S. Katar, and L. Yonge, "Homeplug 1.0 powerline communication LANs - protocol description and performance results," *Int. Journal of Communication Systems*, vol. 16, no. 5, pp. 447 – 4732, May 2003.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993.

[5] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[6] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[7] M. Lentmaier and K. S. Zigangirov, "Iterative decoding of generalized low-density parity-check codes," in *Proc. IEEE Int. Symp. on Inf. Theory*, Boston, USA, Aug. 1998, p. 149.

[8] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *Proc. IEEE Int. Conf. on Communications*, vol. 1, Vancouver, Canada, June 1999, pp. 441–445.

[9] D. Rankin and T. Gulliver, "Single parity check product codes," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1354–1362, Aug. 2001.

[10] F. Chiaraluce and R. Garello, "Extended Hamming product codes analytical performance evaluation for low error rate applications," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2353–2361, Nov. 2004.

[11] R. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[12] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[13] L. Tolhuizen, C. Baggen, and E. H. Nowacka, "Union bounds on the performance of product codes," in *Proc. IEEE Int. Symp. on Inf. Theory*, Boston, MA, USA, Aug. 1998, p. 267.

[14] L. M. Tolhuizen, "More results on the weight enumerator of product codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2573–2577, Sept. 2002.

[15] M. El-Khamy and R. Garello, "On the weight enumerator and the maximum likelihood performance of linear product codes," *IEEE Trans. Inf. Theory, submitted*, Dec. 2005.

[16] F. Mac Williams and N. Sloane, *The theory of error-correcting codes*. North Holland Mathematical Libray, 1977, vol. 16.

[17] F. Chiaraluce and R. Garello, "On the asymptotic performance of Hamming product codes," in *Proc. 6th Int. Symp. on Communication Theory and Applications*, Ambleside, UK, July 2001, pp. 329–334.

[18] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[19] L. C. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1698–1709, Nov. 1996.

[20] C. Berrou, R. Pyndiah, P. Adde, C. Douillard, and R. L. Bidan, "An overview of turbo codes and their applications," in *Proc. European Conf. Wireless Technology*, Paris, France, Oct. 2005, pp. 1–9.

[21] D. Williams, "Turbo product code tutorial," in *Open Forum Tutorials at 802.16 Interim Meetings*, May 2000. [Online]. Available: http://www.ieee802.org/16/tutorial

[22] J. Cuevas, P. Adde, S. Kroudan, and R. Pyndiah, "New architecture for high rate turbo decoding of product codes," in *Proc. IEEE Global Telecom. Conf.*, Taipei, China, Nov. 2002, pp. 1363–1367.

[23] G. Zemor, "On expander codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 835–837, Feb. 2001.

[24] A. Barg and G. Zemor, "Distance properties of expander codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 78–90, Jan. 2006.

[25] J. Chen and R. Tanner, "A hybrid coding scheme for the Gilbert-Elliott channel," *IEEE Trans. Commun.*, vol. 54, no. 10, pp. 1787–1796, Oct. 2006.

[26] Y. Wang and M. Fossorier, "Doubly generalized LDPC codes over the AWGN channel," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 899–907, May 2009.

[27] E. Paolini, M. Fossorier, and M. Chiani, "Doubly-generalized LDPC codes: Stability bound over the BEC," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1027–1046, Mar. 2009.

[28] ——, "Generalized and doubly-generalized LDPC codes with random component codes for the binary erasure channel," *IEEE Trans. Inf. Theory*, to appear in April 2010.

[29] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL INP, Tech. Rep., Aug. 2003, 42-154.

[30] G. Liva, W. E. Ryan, and M. Chiani, "Quasi-cyclic Generalized LDPC codes with low error floors," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 49–57, Jan. 2008.

[31] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[32] D. Divsalar, "A simple tight bound on error probability of block codes with application to turbo codes," JPL TMO Progress Report, Tech. Rep., Nov. 1999.

[33] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[34] A. Orlitsky, K. Viswanathan, and J. Zhang, "Stopping set distribution of LDPC code ensembles," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 929–953, Mar. 2005.

[35] M. Flanagan, E. Paolini, M. Chiani, and M. P. C. Fossorier, "Growth rate of the weight distribution of doubly-generalized LDPC codes: General case and efficient evaluation," in *Proc. IEEE Global Telecom. Conf.*, Honolulu, HI, USA, Nov./Dec. 2009.

[36] E. Paolini, M. Flanagan, M. Chiani, and M. P. C. Fossorier, "Spectral shape of check-hybrid GLDPC codes," in *Proc. IEEE Int. Conf. on Communications*, Cape Town, South Africa, May 2010.

[37] M. Flanagan, E. Paolini, M. Chiani, and M. P. C. Fossorier, "On the growth rate of the weight distribution of irregular doubly-generalized LDPC codes," *IEEE Trans. Inf. Theory*, submitted.

[38] S. Abu-Surra, W. E. Ryan, and D. Divsalar, "Ensemble enumerators for protograph-based generalized LDPC codes," in *Proc. IEEE Global Telecom. Conf.*, Washington, DC, USA, Nov. 2007, pp. 1492–1497.

[39] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

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

[41] M. Lentmaier, D. V. Truhachev, and K. S. Zigangirov, "Iteratively decodable sliding codes on graphs," in *Proc. 8th Int. Workshop on Algebraic and Combinatorial Coding Theory*, St. Petersburg, Russia, Sept. 2002, pp. 190–193.

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

[43] 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.

[44] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.