Non-binary Decoding of Structured LDPC Codes: Density Evolution

Daniele Capirone, Giacomo Como, Fabio Fagnani and Federica Garin
Politecnico di Torino, C.so Duca degli Abruzzi 24, 10129 Torino, Italy
Email: daniele.capirone@polito.it, giacomo.como@polito.it, fabio.fagnani@polito.it, federica.garin@polito.it

***Abstract*—A class of serial turbo codes admitting low-density parity-check (LDPC) representation is considered. Their parity matrix has a random and a structured part. Thanks to their turbo structure, these codes are linear-time encodable, while they can be decoded as LDPC codes.**

Previous works enlightened the role of the inner encoder in the error floor region and suggested the use of a non-binary iterative decoding algorithm. In this paper, a density-evolution analysis is developed giving insight into the performance of these codes in the waterfall region. The inner encoder is optimized in order to guarantee the best tradeoff between error floor and threshold.

I. Introduction

One of the main problems of LDPC codes is their encoding complexity, which is generally quadratic in the block length, as the generating matrix is not low density. This issue has been addressed in two different ways. On one side there are the results in [9], which allow to construct, for given generic LDPC matrices, equivalent generating matrices with lower encoding complexity. On the other side, constraining the parity check matrix to have a particular structure can a priori guarantee easy encoding. A successful construction uses matrices with a staircase part (i.e. a sub-matrix with ones on the diagonal and on the lower diagonal, and zeros everywhere else), so that the encoder can be seen as the serial concatenation of a repetition code, an interleaver and an accumulator. They are called Repeat-Accumulate (RA) [5] codes or, if repetition is not uniform, Irregular Repeat-Accumulate (IRA) codes [4].

We follow this second approach, studying LDPC codes encodable with a serial turbo structure. There is a wide literature on the analysis and design of IRA codes (we refer to [10] and references therein). Previous works [3] and [2] investigate the possibility to vary the structured part of the parity matrix, which is equivalent to replacing the accumulator by another convolutional inner encoder.

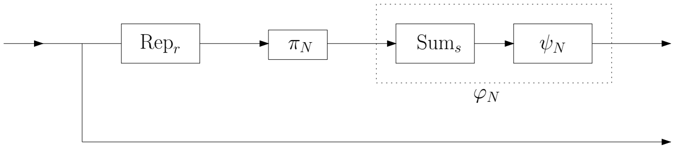
Theoretical results in [3] predict that performance in the error floor region can be significantly improved by optimizing the inner encoder. Such an analysis, assuming ML decoding, was confirmed in [2] using a non-binary decoding algorithm that hides the small cycles present in the standard Tanner graph. MonteCarlo simulations showed that hierarchies indicated by the ML analysis were respected in the medium-high SNR region.

In this paper, we turn our attention to the behaviour of these structured codes in the waterfall region. Inspired by [10] and [7], we use density evolution to analyze the non-binary decoding algorithm proposed in [2]. This relates the threshold to the choice of the inner encoder. It is observed that Repeat-Accumulate exhibits the best threshold, and the worst error floor behaviour. On the other hand, we show that it is possible to design structured LDPC codes with optimal error-floor behaviour and good threshold.

Simulations both on BEC and on AWGN channel validate our analysis.

II. Coding Scheme and Decoding Algorithm

Consider the family of Repeat-Sum-Convolute (RSC) codes, defined by the following encoding structure:



By $\mathrm{Rep}_r : \mathbb{F}_2^N \to \mathbb{F}_2^{rN}$ we denote the repetition code with rate $1/r$; $\mathrm{Sum}_s : \mathbb{F}_2^{rN} \to \mathbb{F}_2^{rN/s}$ is defined by

$$\mathrm{Sum}_s(\boldsymbol{x}) = (x_1 + \ldots + x_s,\, x_{s+1} + \ldots + x_{2s},\, \ldots)$$

i.e. it gives the modulo-2 sum of every block of s bits. Finally, let $\psi(D) : \mathbb{F}_2^k((D)) \to \mathbb{F}_2^k((D))$ be a rate-1 non-catastrophic and recursive convolutional encoder, and $\psi_N : \mathbb{F}_2^{rN/s} \to \mathbb{F}_2^{rN/s}$ be the truncated encoder obtained by using the trellis of $\psi(D)$ for $rN/(sk)$ time steps. Define the rate $R = (1 + \frac{r}{s})^{-1}$ systematic encoder

$$\Phi_N : \mathbb{F}_2^N \to \mathbb{F}_2^{(1+\frac{r}{s})N}, \qquad \Phi_N \boldsymbol{u} = (\boldsymbol{u}, \mathrm{Sum}_s \circ \pi_N \circ \mathrm{Rep}_r\, \boldsymbol{u})\,.$$

We will always assume that rN is a multiple of sk, so that the above construction can be properly made.

Φ_N is a particular kind of systematic serial turbo encoder where the outer encoder is Rep_r and the inner encoder is $\varphi_N = \psi_N \circ \mathrm{Sum}_s$. φ_N can be considered as the truncation of a proper convolutional encoder, which is not injective, but the transmission of the systematic bits ensures injectivity and non-catastrophicity of Φ_N.

The representation as serial turbo codes allows an encoding time linear with kN. The decoding can be performed exploiting the fact that these codes have a natural LDPC representation.

Indeed, notice that a pair $(\boldsymbol{u}, \boldsymbol{c})$ in $\mathbb{F}_2^N \times \mathbb{F}_2^{rN/s}$ is in the image of Φ_N if and only if $\boldsymbol{c} = \psi_N \circ \mathrm{Sum}_s \circ \pi_N \circ \mathrm{Rep}_r(\boldsymbol{u})$.

This is equivalent to $\mathrm{Sum}_s \circ \pi_N \circ \mathrm{Rep}_r(\boldsymbol{u}) + \psi_N^{-1}(\boldsymbol{c}) = \mathbf{0}$ and can be represented in the matrix form $[H_N\, K_N]\left[\begin{smallmatrix}\boldsymbol{u}\\ \boldsymbol{c}\end{smallmatrix}\right] = \mathbf{0}$. Here H_N is a $\frac{r}{s}N \times N$ matrix depending on the permutation π_N only. It is sparse, having at most s ones per row and r ones per column. K_N is a $\frac{r}{s}N \times \frac{r}{s}N$ matrix depending on the choice of ψ only. It is also low density, having a number of ones per row and per column bounded by $k(\deg \psi^{-1}(D)+1)$.

In [3], RSC ensembles are analyzed with classical tools from turbo codes literature [1]: an interleaver gain is proved (at medium-high SNR), in the sense that ML bit and word error probabilities, averaged over all interleavers, are asymptotically vanishing when the interleaver length grows to infinity, provided that $r \geq 2$ and $r \geq 3$ respectively. The same paper also provides a design parameter for the inner encoder: the error floor can be lowered by increasing d_2^{ψ}, i.e. the minimum Hamming weight of codewords of $\psi(D)$ corresponding to input weight two. This result is obtained studying a smaller ensemble where some spread is enforced in the interleaver (or equivalently 2-cycles are forbidden from the Tanner graph associated with the parity-check matrix).

Decoding can be done by message passing on the Tanner graph associated with the LDPC matrix. However, simulation results with this algorithm do not match with the theoretical predictions in [3]. For example, most codes, even with high d_2^{ψ}, do not outperform the simple accumulator. The poor performance of such codes can be explained by the presence of $O(N)$ small cycles in the structured part of the Tanner graph, as opposed to the staircase, cycle-free structured K_N of the accumulator.

In this paper, we focus on a specific family of RSC codes, where the inner encoder has the form $\psi(D) = (A + BD)^{-1}$ with $A, B \in \mathbb{F}_2^{k\times k}$ invertible matrices. Note that any non-catastrophic convolutional encoder $\psi(D)$ can be represented in this form, with possibly non-invertible A and B; the invertibility assumption could be relaxed, but here, for the sake of simplicity, we don't deal with the most general case.

The form $(A + BD)^{-1}$ corresponds to a structured part of the parity matrix which is block-wise staircase:

$$K_N = \begin{bmatrix} A & 0 & 0 & \dots & 0 \\ B & A & 0 & \dots & 0 \\ 0 & B & A & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & B & A \end{bmatrix}$$

Obviously if $k = 1$ and $A = B = 1$ this reduces to the accumulator. This structure can be exploited by a non-binary algorithm proposed in [2], which avoids cycles.

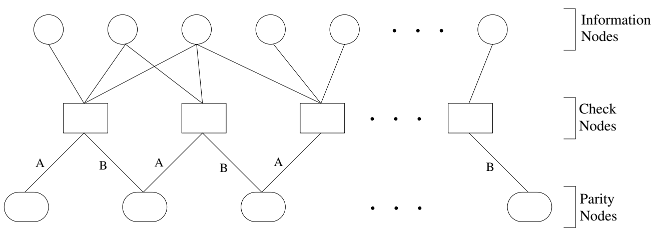


Fig. 1. Tanner Graph of the hybrid nonbinary algorithm

It applies a sum-product belief propagation algorithm on the graph represented in Fig. 1. This associates to the parity matrix $[H_N K_N]$ a labeled factor graph with vertex set given by $\mathcal{V}_i \cup \mathcal{V}_p \cup \mathcal{V}_c$, where:

- $\mathcal{V}_i = \{i_1, \dots, i_N\}$ is a set of N information nodes, each corresponding to an information bit (recall the codes are systematic);
- $\mathcal{V}_p = \{p_1, \dots, p_{\frac{r}{ks}N}\}$ is a set of $\frac{r}{ks}N$ parity nodes, each corresponding to a group of k consecutive parity bits;
- $\mathcal{V}_c = \{c_1, \dots, c_{\frac{r}{ks}N}\}$ is a set of $\frac{r}{ks}N$ check nodes each corresponding to a group of k consecutive rows of the matrix.

For every $1 \leq j \leq \frac{r}{ks}N$, the parity node p_j is connected only to the check node c_j with an edge labeled by $\lambda_{i_j,c_j} = A$, and to the check node c_{j+1} with an edge labeled by $\lambda_{i_j,c_{j+1}} = B$. There is an edge between a check node c_l in $\mathcal{V}_c$ and an information node i_j in $\mathcal{V}_i$ whenever the $k \times 1$ block $(H_N)_{[k(l-1)+1,kl],j}$ is nonzero; such an edge is labeled by the $k \times 1$ block $\lambda_{c_l,p_j} = (H_N)_{[k(l-1)+1,kl],j}$ itself.

Messages exchanged between information nodes and check nodes lay in $\mathcal{P}(\mathbb{F}_2)$ while messages exchanged between parity nodes and check nodes lay in $\mathcal{P}(\mathbb{F}_2^k)$ (i.e. probability distributions over $\mathbb{F}_2$ and $\mathbb{F}_2^k$ respectively).

Denote the message from node v to node v' at time t by $\boldsymbol{m}^t_{v\to v'}$. For every adjacent parity node v and check node c initialize $\boldsymbol{m}^0_{c\to v}$ as the uniform distribution over $\mathbb{Z}_2^k$ and similarly for every adjacent information node v and check node c let $\boldsymbol{m}^0_{c\to v}$ be the uniform distribution over $\mathbb{Z}_2$. Then for every time step $t \geq 1$

- the message sent from a node v in $\mathcal{V}_i \cup \mathcal{V}_p$ to an adjacent check node c, $\boldsymbol{m}^t_{v\to c}$ is the normalized pointwise product of $\boldsymbol{z}_v$, the channel output message at time v, and of messages $\boldsymbol{m}^{t-1}_{c'\to v}$ received by the node v from all its neighbors c' but c;
- the message sent from a check node c to an adjacent information or parity node v is given by
$$\boldsymbol{m}^t_{c\to v}(x) = \mathbb{P}^t_{c\to v}\big(\textstyle\sum_{\substack{v'\sim c\\ v'\neq v}} \lambda_{c,v'}X_{v'} = \lambda_{cv}x\big)$$
where the probability $\mathbb{P}^t_{c\to v}$ is evaluated by considering the random variables $X_{v'}$ mutually independent, each distributed according to $\boldsymbol{m}_{v'\to c}$.

III. Density Evolution

We analyze the previously described algorithm using the density evolution (DE) method [8]. The key remark is that this approach is possible because the Tanner graph corresponding to the non-binary decoding algorithm does not present cycles in the structured part, and classical results on regular LDPC ensembles [8] ensure that the random part is locally tree-like with high probability. So, in addition to a possible improvement in performances, the non-binary algorithm also makes possible an analysis which was not allowed in general for RSC ensambles with binary message-passing, because of the structural presence of short cycles in the Tanner graph.

Consider a family of binary-input output-symmetric (BIOS) channels ordered by physical degradation and indexed by a parameter ϵ. Then, asymptotically in N, the behaviour of the algorithm can be characterized by a dynamical system, which assumes the following form, reflecting the structure of the Tanner graph of Fig. 1:

$$\begin{cases} \boldsymbol{y}_{t+1} = f_\epsilon(\boldsymbol{y}_t, \boldsymbol{x}_t^A, \boldsymbol{x}_t^B) \\ \boldsymbol{x}_{t+1}^A = f_\epsilon^A(\boldsymbol{x}_t^A, \boldsymbol{y}_t) \\ \boldsymbol{x}_{t+1}^B = f_\epsilon^B(\boldsymbol{x}_t^B, \boldsymbol{y}_t) \end{cases} \tag{1}$$

In (1), for all $t \geq 0$, $\boldsymbol{y}_t \in \mathcal{P}(\mathcal{P}(\mathbb{F}_2))$ is the density of messages sent at time t from information nodes, while $\boldsymbol{x}_t^A$ and $\boldsymbol{x}_t^B$ in $\mathcal{P}(\mathcal{P}(\mathbb{F}_2^k))$ are the densities of messages sent by parity nodes, on edges labeled by A and B respectively (under the assumption that there is no cycle of length smaller than $4t$). Note that for parity nodes we need to consider separately the densities of messages on edges with label A and B, because we have fixed matrices A, B and we cannot use the simplification given by averaging. We can exploit the averaging effect only for information nodes, which have random labels on their output edges.

A. Density evolution on BEC

While for general BIOS channels (as the AWGN or the BSC) the dynamical system (1) is infinite-dimensional, we will focus here on the BEC, where it reduces to finite dimension. In this case, assuming transmission of the all-zero codeword (assumption justified by symmetry of the channel and code linearity), the messages sent during the decoding are:

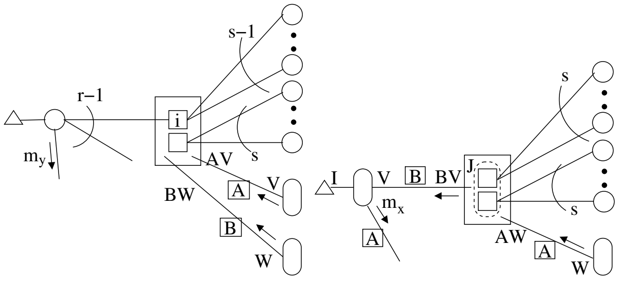
- from and to an information node: either 0 or 'erased'. (i.e. the uniform probability on the set $\{0\}$ and $\{0, 1\}$ respectively);
- from and to a parity node: the uniform distribution on some vector subspace of $\mathbb{F}_2^k$, possibly $\{0\}$ or $\mathbb{F}_2^k$ itself. If the message comes from the channel, not all subspaces are possible, only those corresponding exactly to the restriction of $\mathbb{F}_2^k$ to some of its components, i.e. the spaces having as a basis a subset of the canonical basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k\}$ of $\mathbb{F}_2^k$.

More simply, one can identify the messages with the subspaces themselves, so that the alphabet of messages from and to the parity nodes is $G := \{$ subspaces of $\mathbb{F}_2^k$ $\}$.

So, the density evolution system will have the following variables:

- $y_t \in [0, 1]$ = fraction of information bits erased at time t;
- $\boldsymbol{x}_t^A \in \mathcal{P}(G)$ defined by $x_t^A(V)$ = fraction of output messages from parity symbols (on edge with label A), which at time t are equal to V;
- analogous definition for $\boldsymbol{x}_t^B$ on edges with label B.

We will use the short-hand notations $[k] := \{1, 2, \ldots, k\}$ and, for $I \subseteq [k]$, $\mathbb{F}_2^I := \operatorname{span}\{\boldsymbol{e}_i, i \in I\}$. We will denote by $\pi_i(V)$ the restriction of a vector space $V \in G$ to its i-th component, i.e. $\pi_i(V)$ is $\{0\}$ if all vectors in V have their i-th component equal to zero, and is $\{0, 1\}$ otherwise.



(a) Information Node (b) Parity Node

Fig. 2. Portions of Tanner graph, with messages exchanged

With these notations, the equations describing the density evolution on the BEC are:

$$y_{t+1} = \epsilon \left[1 - \frac{(1 - y_t)^{s-1}}{k} \sum_{i=1}^{k} \sum_{\substack{V \in G: \\ \pi_i(AV) = \{0\}}} x_t^A(V) \sum_{\substack{W \in G: \\ \pi_i(BW) = \{0\}}} x_t^B(W) \right]^{r-1} \tag{2}$$

and, for any $U \in G$,

$$x_{t+1}^A(U) = \sum_{I \subseteq [k]} \sum_{J \subseteq [k]} \sum_{W \in G} \epsilon^{|I|} (1 - \epsilon)^{k - |I|} x_t^A(W)\, p_J(y_t) n_{U,W,I,J}^A \tag{3}$$

where:

- $p_J(y_t) = (1 - (1 - y_t)^s)^{|J|} (1 - y_t)^{s(k - |J|)}$
- $n_{U,W,I,J}^A = \left| \{ V \in G : BV = AW + \mathbb{F}_2^J,\, U = V \cap \mathbb{F}_2^I \} \right|$; using the assumption that B is invertible, $n_{U,W,I,J}^A$ becomes simply 1 if $U = B^{-1}(AW + \mathbb{F}_2^J) \cap \mathbb{F}_2^I$ and 0 otherwise.

The equation for $\boldsymbol{x}_{t+1}^B$ is the same as (3), simply exchanging the role of A and B everywhere.

Fig. 2 helps to understand the meaning of equations (2) and (3). It shows the portion of Tanner graph corresponding to one updating step in the iterative decoding, from the perspective of an information and a parity node. The triangles denote the output from the channel. The check nodes can be thought as the aggregation of k bit-wise check nodes. The i-th bit-wise check is connected to the information nodes having label $\boldsymbol{e}_i$.

Referring to Fig. 2(a), y_{t+1} is the probability that $m_y = \{0, 1\}$. This happens only if both the message from the channel and all the $r - 1$ incoming messages from check nodes give an erasure. The channel sends an erasure with probability ϵ. For each of the $r - 1$ edges, we will now compute the probability that the message is $\{0\}$, i.e. not erased, assuming that the label is $\boldsymbol{e}_i$: the averaging on i then comes from the fact that the labels are uniformly random. Looking at the check node, we see that it sends $\{0\}$ for the i-th component when all the other $s - 1$ edges incoming with label $\boldsymbol{e}_i$ carry a $\{0\}$ and both messages AV and BW from parity nodes give $\{0\}$ when restricted to the i-th component. This happens with probability $(1 - y_t)^{s-1} \sum_{V : \pi_i(AV) = \{0\}} x_t^A(V) \sum_{W : \pi_i(BW) = \{0\}} x_t^B(W)$.

$\boldsymbol{x}_{t+1}^A$ is the distribution of the messages m_x in Fig. 2(b); let's compute the probability that $m_x = U$. Note that U is the intersection of the message received from the channel and the one coming from the check node. The channel can send any of the spaces $\mathbb{F}_2^I$, $I \subseteq [k]$ (i.e. an erasure exactly in the components listed in the indexes set I), each with probability $\epsilon^{|I|}(1-\epsilon)^{k-|I|}$. The check node computes the sum of the vector spaces it receives. The combination of the ks messages from the information nodes is $\mathbb{F}_2^J$, $J \subseteq [k]$, i.e. an erasure exactly in the components listed in J, with probability $p_J(y_t)$. In fact, a bit-wise check node j is erased when at least one of the s information nodes with label $\boldsymbol{e}_j$ carries an erasure. The check node has to sum $\mathbb{F}_2^J$ and the message it receives on the edge labeled with A, which is AW with probability $x_t^A(W)$, for any $W \in G$. In conclusion, any triple $I \subseteq [k]$, $J \subseteq [k]$, and $W \in G$ appears with probability $\epsilon^{|I|}(1-\epsilon)^{k-|I|}p_J(y_t)x_t^A(W)$ and contributes to $x_{t+1}^A(U)$ if and only if $U = B^{-1}(AW + \mathbb{F}_2^J) \cap \mathbb{F}_2^I$, i.e. $n^A_{U,W,I,J} \neq 0$.

The evolution equations (2) and (3) describe a dynamical system, with variable $\boldsymbol{z} = (y, \boldsymbol{x}^A, \boldsymbol{x}^B) \in [0,1] \times \mathcal{P}(G) \times \mathcal{P}(G)$. It is clear that if we denote by δ_V a vector in $\mathcal{P}(G)$ with a one in position V and zeros everywhere else, $\boldsymbol{z}^* := (0, \delta_{\{0\}}, \delta_{\{0\}})$ is a fixed point of the system. Since $y_t \to 0$ represents successful decoding, finding the threshold means finding up to what value of ϵ the system converges to $\boldsymbol{z}^*$ from the initial condition $\boldsymbol{z}_0 = (1, \delta_{\mathbb{F}_2^k}, \delta_{\mathbb{F}_2^k})$. This choice of $\boldsymbol{z}_0$ corresponds to the initialization of the decoding algorithm.

Numerical computation of the threshold for different values of A and B can guide the choice of the inner encoder, as is discussed in Section IV.

An interesting theoretical question which is often considered in the Density Evolution literature is the stability condition: you look for conditions ensuring that the fixed point to which you wish convergence (in our case, $\boldsymbol{z}^*$) is asymptotically stable for all ϵ, i.e. for all ϵ, there exists a neighbourhood of $\boldsymbol{z}^*$ such that starting from any initial condition in that neighbourhood the system will converge to $\boldsymbol{z}^*$. This is clearly a necessary condition for convergence from your given initial condition, and it can provide interesting design guidelines, as it happens for the degree distributions of the irregular binary random LDPC ensemble.

In our setting, it turns out that $\boldsymbol{z}^*$ is asymptotically stable, for all ϵ, for any choice of A and B, provided that $r \geq 3$. This generalizes the well-known result that for the regular LDPC ensemble, with left degree at least three, the asymptotic stability of the fixed point 0 is always true. However, the proof in our setting is less trivial. You need at first to linearize the system, i.e. to compute the Jacobian matrix in $\boldsymbol{z}^*$, $J(\boldsymbol{z}^*)$. $r \geq 3$ ensures that the first line of $J(\boldsymbol{z}^*)$ is all-zero. Then you can note that $\boldsymbol{x}_{t+1}^A$ does not depend on $\boldsymbol{x}_t^B$ and depends linearly on $\boldsymbol{x}_t^A$; denote by M_A the $|G| \times |G|$ matrix describing this linear map in the case when $y_t = 0$. Analogously define M_B. Now note that the eigenvalues of $J(z^*)$ are: 0 and then the eigenvalues of M_A and M_B. Now, instead of explicitly computing the eigenvalues of A and B, which are hard to express in closed form, we prove that the linear systems on $\mathcal{P}(G)$ associated with M_A and M_B have a unique asymptotically stable fixed point in $\delta_{\{0\}}$, by using a Lyapunov technique (see e.g. [6]): we define the function $\eta(x) = \sum_{U \in G} (\dim U) x(U)$, which can be interpreted as the average dimension of the subspaces of $\mathbb{F}_2^k$ with respect to the probability distribution $\boldsymbol{x}$. We note that η is a linear function, $\eta(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathcal{P}(G)$, $\eta(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = \delta_{\{0\}}$, and we prove that η is strictly decreasing along the trajectories, i.e. $\eta(M_A\boldsymbol{x}) < \eta(\boldsymbol{x})$ and $\eta(M_B\boldsymbol{x}) < \eta(\boldsymbol{x})$ for all $\boldsymbol{x} \neq \delta_{\{0\}}$.

IV. Simulations and Numerical Results

Our analysis is validated by simulation results in which at low SNR the hierarchy given by the threshold is respected. The threshold has been obtained numerically, iteratively calculating message densities and considering a maximum of 250 iterations.

All the examples simulated have $r = 4$ and $s = 4$, so that the overall rate R is $1/2$. Simulations differ for k and for the choice of $\psi(D)$, which influences both the threshold and the parameter d_2^ψ.

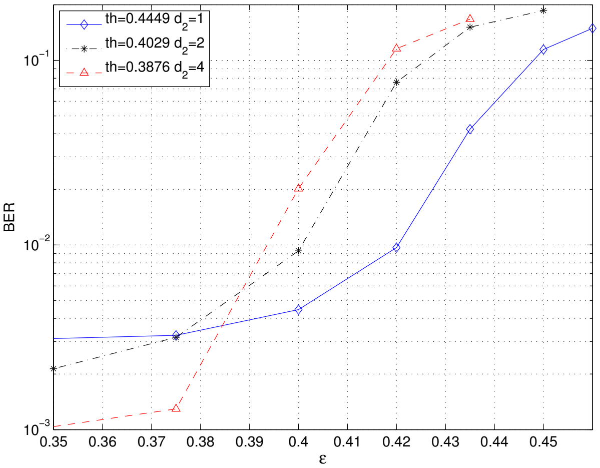


Fig. 3. Results on BEC channel, $k = 3$, block length 2400, rate 1/2

Fig. 3 shows the behaviour of the non-binary decoding algorithm with $k = 3$ for three encoders, on the BEC channel. For the low SNR region the predicted hierarchies are respected, they can be read on the graph in the BER region between 10^{-1} and 10^{-2}. Hierarchies are reversed at higher SNR, as predicted by the parameter d_2^ψ.

Fig. 4 shows analogous simulations for some codes with $k = 4$. For all the curves the matrix B has been kept fixed equal to the identity while A, starting from the identity matrix, has been filled up with more and more ones: this leads to a decreasing threshold and a d_2^ψ which is very low when A is sparse; this suggests some relation between our design parameters and sparseness of the matrices. Simulation results are again perfectly matching with the predictions.

Simulations show that performance on the AWGN respects the same hierarchies of the BEC thresholds; see e.g. Fig. 5. This suggests that density evolution on the BEC can also provide some insight into the behaviour on other channels, and give guidelines for the choice of the encoding scheme. Another possibility, which will be considered in future work,

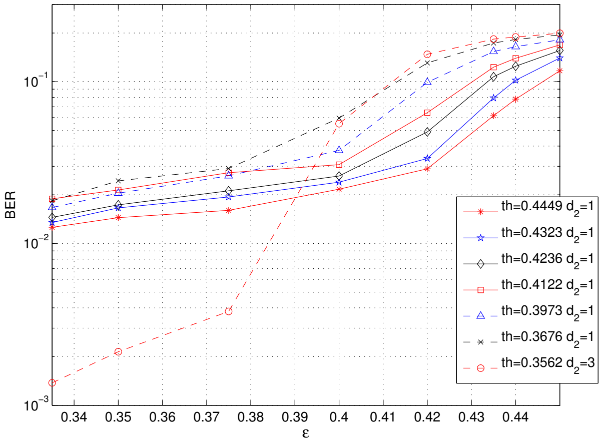


Fig. 4. Results on BEC channel, $k = 4$,, block length 2000, rate 1/2

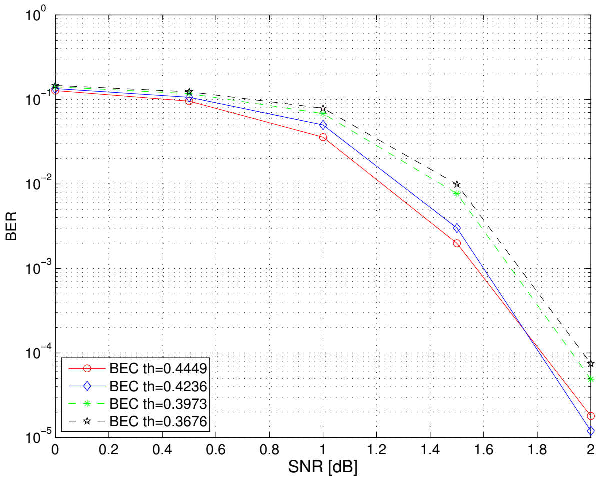


Fig. 5. Results on AWGN channel, $k = 4$, block length 2000, rate 1/2

is to study finite-dimensional approximations of the AWGN density evolution, as it is done for IRA codes in [10].

In [3], we have underlined the role of the parameter d_2^{ψ}: its maximization improves performance at high SNR. Now density evolution provides an optimization criterion for low SNR: maximizing the threshold. It is well-known that these two optimizations are often in contrast, so that a compromise is necessary if both SNR regions are targeted. We want to investigate if this happens also in our setting.

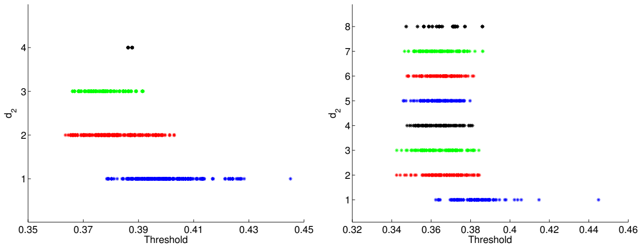


Fig. 6. Distribution of the threshold as function of d_2^{ψ} with $k = 3, 4$

Fig. 6 reports the threshold vs. d_2^{ψ} for a large number of choices of the pairs A, B. These numerical results show that the best threshold corresponds to A and B being both permutation matrices; unfortunately, it is easy to prove that this condition implies $d_2^{\psi} = 1$, the same as with the simple accumulator on which we wanted to improve. For $k = 3$ we see that the values of the threshold are quite dispersed, even if there is some dependence on d_2^{ψ}, in that the lower d_2^{ψ}, the higher the maximum threshold. For $k = 4$, apart from the special case $d_2^{\psi} = 1$, the thresholds don't exhibit any apparent dependence on d_2^{ψ}. This suggest, especially for $k = 4$, the following simple design criterion: take A and B with the maximum threshold among those having the maximum d_2^{ψ}.

V. Conclusion and Future Works

This paper has analyzed the behaviour of the non-binary decoding algorithm presented in [2], applied to a family of linear-time encodable LDPC codes. We have developed the density evolution that allows the prediction of the performance for the low-medium SNR region, on the BEC. MonteCarlo simulations have confirmed the theoretical and numerical results both on BEC and AWGN channels. The density evolution has showed to be a powerful tool for optimizing such a family of codes and it can be jointly used with the effective free distance of the inner encoder (d_2^{ψ}), which influences the error floor.

Further investigations will be devoted to the case of irregular degrees in the random part of the Tanner graph (equivalent to time-varying repetition and summation codes). Re-writing equations (2) and (3) for the irregular case is straightforward, while some theoretical analysis, even just the stability condition, looks a hard task. We plan to use numerical values of the threshold as a tool for optimizing irregular degrees distribution, as is usually done in the unstructured case.

Acknowledgment

The authors would like to gratefully acknowledge Roberto Garello for having encouraged and supported this work.

References

[1] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding”, *IEEE Trans. on Inf. Th.*, vol. 44, pp. 909–926, May 1998.

[2] D. Capirone, G. Como, F. Fagnani and F. Garin, “Nonbinary decoding of structured LDPC codes”, *IZS 2008*.

[3] F. Garin, G. Como and F. Fagnani, “Staircase and other structured linear-time encodable LDPC codes: analysis and design”, *Proc. ISIT 2007*.

[4] H. Jin, A. Khandekar and R. J. McEliece, “Irregular Repeat-Accumulate Codes”, *Proc. of the 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, Sept. 2000.

[5] H. Jin and R. J. McEliece, “Coding theorems for turbo code ensembles”, *IEEE Trans. on Inform. Theory*, vol. 48 (6), pp. 1451–1461, June 2002.

[6] J. P. LaSalle, *The Stability and Control of Discrete Processes*, Applied Mathematical Sciences, 62, SpringerVerlag, 1986.

[7] V. Rathi and R. Urbanke, “Density evolution, thresholds and the stability condition for non-binary LDPC codes”, *IEE Proceedings-Communications*, vol. 152 (6), pp. 1069–1074, Dec 2005.

[8] T. J. Richardson and R. Urbanke, “The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding”, *IEEE Trans. Inform. Theory*, vol. 47 (2), pp. 599-618, Feb. 2001.

[9] T. J. Richardson and R. Urbanke, “Efficient Encoding of Low-Density Parity-Check Codes”, *IEEE Trans. on Inform. Theory*, vol. 47 (2), pp. 638–656, Feb. 2001.

[10] A. Roumy, S. Guemghar, G. Caire and S. Verdú, “Design Methods for Irregular Repeat-Accumulate Codes”, *IEEE Trans. on Inform. Theory*, vol. 50 (8), Aug. 2005.