

Does the Performance of LDPC Codes depend on the Channel?

Michele Franceschini, Gianluigi Ferrari and Riccardo Raheli

Università di Parma
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181A
I-43100, Parma, Italy
E-mail: mfrance@tlc.unipr.it
gianluigi.ferrari@unipr.it
raheli@unipr.it

Abstract

In this paper, we analyze the performance of Low Density Parity Check (LDPC) codes on memoryless channels. We use a recently proposed analysis technique for iterative decoding based on EXtrinsic Information Transfer (EXIT) charts. We show that, based on this technique, the predicted performance of an LDPC code does not depend on the specific memoryless channel, but only on the mutual information (MI) between the input and the output of the channel. As a validation of this conjecture, we evaluate the performance of some LDPC codes over five representative memoryless channels and we compare them, obtaining results in excellent agreement with our conjecture.

1. Introduction

In the last decade, the introduction of new coding techniques has made it possible to achieve near-capacity transmission over the additive white Gaussian noise (AWGN) channel. In particular, the codes invented in [1], i.e., Low Density Parity Check (LDPC) codes, were recently rediscovered and analyzed in [2]. The assumptions considered in [1] for the derivation of an iterative decoding algorithm are relative to a *memoryless* channel, i.e., a channel such that, conditionally on the transmitted symbols, its output observations are mutually independent. *Regular* LDPC codes, as introduced in [1], have been extended to *irregular* LDPC codes which show improved performance [3]. In particular, an ensemble of LDPC codes can be defined in terms of *degree distributions* [4], which correspond to polynomials describing (statistically) the internal structure of the codes. In [5], a powerful analysis technique, based on EXtrinsic Information Transfer (EXIT) charts and first developed in [6] for Turbo Codes (TC), was applied to LDPC codes.

In this paper, we show that the analysis of LDPC codes based on EXIT charts suggests that the behavior of an *ensemble* of LDPC codes (i.e., a set of codes with given degree distributions) does not depend on the particular memoryless channel, but only on the constrained input capacity (indicated as C_{ci}) of the channel. We then support this conjecture with simulation results relative to various LDPC codes transmitted over the following two groups of memoryless channels [7]. The first group, representative of *symmetric* channels, is constituted by a binary-input (BI)-AWGN channel, a Binary Symmetric Channel (BSC) and a Binary Erasure Channel (BEC); the second group, representative of *asymmetric* channels, is constituted by a Z channel and a Binary Asymmetric Channel (BAC). An early stage version of our conjecture can also be found in [4], where the authors noted that the same codes, optimized for the AWGN channel, showed good performance when used for transmission over other memoryless channels, such as BSC and BEC. We give a comparison of the performance based on the channel parameter C_{ci} , and a simple justification of the obtained results.

The paper outline is as follows. In Section 2, we introduce some tools useful to the subsequent analysis. In Section 3, we introduce the main conjecture underlying this paper. In Section 4, we present some numerical results enforcing our conjecture. Section 5 concludes the paper.

2. LDPC Codes and EXIT Charts

In linear block codes, a vector of bits \mathbf{y} is a codeword iff $H\mathbf{y} = \mathbf{0}$, in which H is a binary matrix defined as *parity check matrix*. LDPC codes are linear block codes with a *sparse* parity check matrix H , i.e., a parity check matrix with a few ones and many zeros [1]. In [1], it is shown how to build, in a one-to-one cor-

rependence with H , a graph consisting of two kinds of nodes, also known as *bipartite graph*. Each node of the first kind, called *check node*, is associated with a row of H ; each node of the second kind, called *variable node*, is associated with a column of H . A *variable node* is connected to a *check node* if there is a “1” in the corresponding position given by the intersection of the respective row and column of H . This interpretation of a linear block code led, in [1], to the design of an asymptotically (in the codeword length) optimal decoding algorithm, such that each node sends and receives real-valued messages through the graph branches. This algorithm has recently been described as a particular instance of a broader class of algorithms also known as *sum-product algorithms* [8].

Following the notation in [4], a node has degree d if it has d branches departing from it. The *degree distributions* of a linear block code correspond to the couple of polynomials $(\lambda(x), \rho(x))$ defined as follows:

$$\rho(x) \triangleq \sum_j \rho_j x^{j-1} \quad (1)$$

$$\lambda(x) \triangleq \sum_i \lambda_i x^{i-1} \quad (2)$$

where ρ_j is the fraction of branches in the graph connected to degree- j *check nodes* and λ_i is the fraction of branches in the graph connected to degree- i *variable nodes*. The polynomial $\rho(x)$ is also known as the *check node degree distribution* and $\lambda(x)$ is also known as the *variable node degree distribution*.

In the decoding process, the nodes on the graph act as “processing blocks.” In particular, the reliability information, generated by a node and transmitted over an edge, depends on the reliability values received from all edges connected to the node [1]. The decoding algorithm is based on the following two basic steps:

1. variable nodes compute and send their messages to the connected check nodes;
2. check nodes compute and send their messages back to the connected variable nodes.

These steps are repeated until a suitable stopping condition is met: for example, a valid codeword has been selected. At the very first iteration, variable nodes compute their messages assuming no information at all from check nodes; this is equivalent to sending channel reliability values, in the form of logarithmic likelihood ratios (LLRs) relative to the codeword symbols, directly to the check nodes.

In the EXIT chart-based analysis of LDPC codes [5, 6, 9], the set of variable nodes and the set of check nodes are treated as a blocks which process *soft* messages at

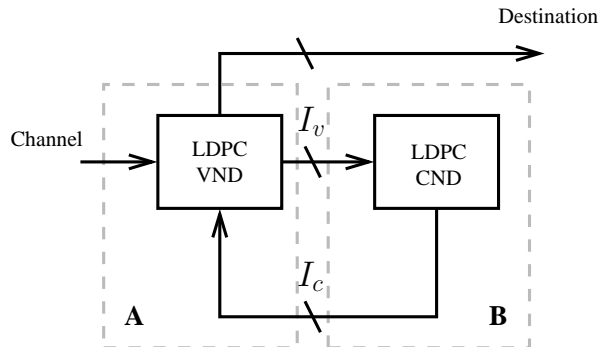


Figure 1: *Block scheme for the receiver.*

their inputs and generate *soft* messages at their output. The ensemble of messages generated by a group of nodes, either variable or check, will be referred to as “message set”. In Figure 1, a pictorial representation of the receiver shows two different blocks: block **A** represents the set of all variable nodes, or variable node detector (VND); block **B** represents the set of all check nodes, or check node detector (CND). Each message set, at the input or output of a block, can be viewed as the output of a channel with the actually transmitted codeword at its input: the message set is indeed a stochastic function of the transmitted codeword. For each message set, the Mutual Information (MI) between the transmitted codeword and the message set can then be computed. In Figure 1, I_v denotes the MI at the output of the VND, and I_c the MI at the output of the CND. The underlying assumption is that the MI at the output of a block is a function of the MI at the input, regardless of the actual statistical distribution of the messages. In [6], sufficient conditions for an accurate analysis are outlined. It can be shown that a simple Gaussian approximation for the messages (seen as random variables) at the output of the variable nodes makes the previous assumption very accurate.

The decoding process can thus be described as a recursive computation of the MI between variable node blocks (LDPC VND) and check node blocks (LDPC CND). Plotting the MI at the output of each block as a function of the MI at the input, i.e., drawing the corresponding EXIT chart, it is possible to predict the convergence characteristics of the decoding process. In Figure 2, the EXIT chart for an LDPC code with $\rho(x) = x^5$ and $\lambda(x) = x^2$ and $E_b/N_0 = 1.8$ dB is shown. The upper function I_v is the MI at the output of variable nodes versus the MI at their input; the lower function I_c^{-1} is the inverse of the MI at the output of check nodes versus the MI at their input. The use of the inverse of the function I_c is expedient for a graphi-

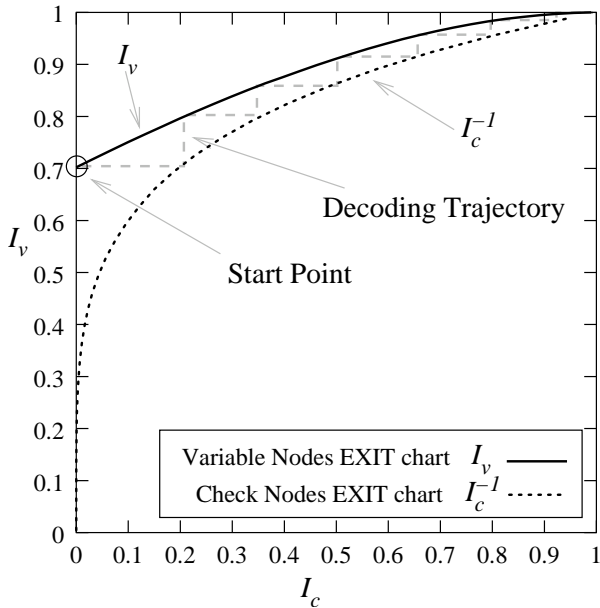


Figure 2: EXIT chart for an LDPC code with $\rho(x) = x^5$ and $\lambda(x) = x^2$. The dashed line represents the trajectory corresponding to the recursive evolution of the MI.

cal study of the recursive evolution of the MI, as shown in Figure 2 by the dashed line. It can be shown [5] that the previous considerations imply that I_v depends on the variable node degree distribution $\lambda(x)$ and on the channel and I_c depends only on the check node degree distribution $\rho(x)$. Note that at the start point of the recursive decoding process, $I_c = 0$, since the check nodes, at the very first iteration, have no information to pass to the variable nodes. The start point of the recursion is therefore $(0, I_v(0))$, as shown in Figure 2.

3. The Start Point in Exit Charts

What is the meaning of the start point in an EXIT chart? The messages at the output of variable nodes at the very first iteration correspond to the LLRs, based on the channel observations, of the transmitted symbols [1]. These quantities are sufficient statistics for an optimal decision on the transmitted sequence. This means that the MI between these LLRs and the transmitted binary sequence is equal to the MI between the channel output and the transmitted binary sequence, also known as the constrained-input channel capacity C_{ci} . Hence, at the first iteration the MI generated at the output of a variable node block in the LDPC decoder is $I_v = C_{ci}$. This value corresponds, as stated in Section 2, to the point $(0, I_v(0))$, i.e. the start point of the EXIT chart decoding trajectory.

In the considered decoding scheme, the channel in-

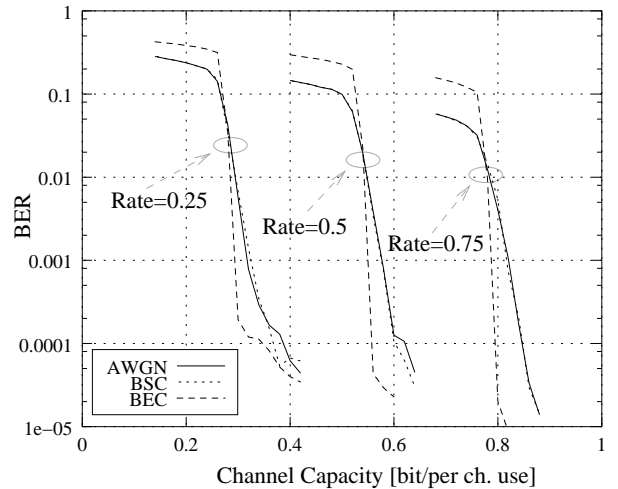


Figure 3: BER versus channel capacity for various LDPC codes transmitted over three binary-input symmetric memoryless channels.

fluences the generation of the LLRs at the variable nodes at the very first iteration. The remaining part of the decoding algorithm is independent on the specific channel. Assuming that the EXIT chart-based analysis is accurate, one could thus conclude that the convergence of the decoding process of an ensemble of LDPC codes, described by their degree distributions, depends only on the constrained input channel capacity and not on the particular channel—provided that the channel is memoryless. This means that if a random code of a given ensemble shows, with high probability, negligible Bit Error Rate (BER) when used over a memoryless channel with given C_{ci} , this code, with high probability, will show negligible BER also when used over any other memoryless channel with equal C_{ci} . This also means that the presence of a processing block between the memoryless channel and the LDPC decoder does not improve the decoding convergence, because it cannot improve the overall channel C_{ci} .

These results are intuitive, and their accuracy is strictly related to the accuracy of the EXIT chart-based analysis. In the next section, simulation results will be presented to support our conjecture.

4. Numerical Results

We have considered Monte Carlo simulations relative to three LDPC codes transmitted over five different memoryless channels. The considered codes, with rates 1/4, 1/2 and 3/4, respectively, were generated starting from optimized degree distributions found in [10]. The codeword length is set to 10000 binary symbols in all cases, and the maximum decoding iteration number has

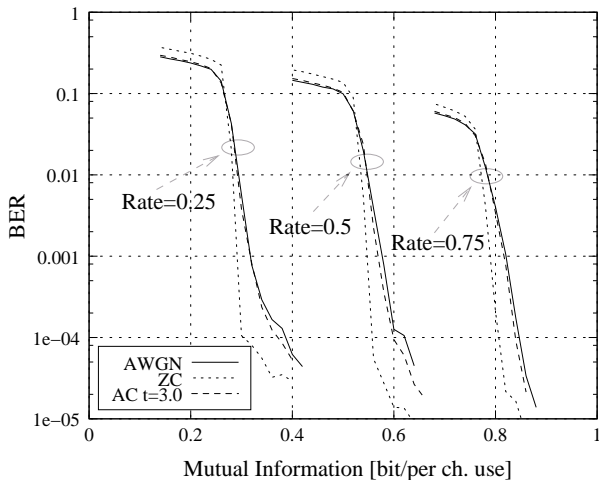


Figure 4: *BER versus channel capacity for various LDPC codes transmitted over two binary-input asymmetric memoryless channels (the AWGN channel is kept for comparison purposes).*

been set to 100 (the decoding process can stop earlier if a codeword is obtained). The chosen memoryless channels are divided in two sets: (i) a binary input AWGN channel, a BSC and a BEC in the first set, and (ii) a Z channel and a BAC in the second set.

All the channels in the first set are symmetric (i.e. channel capacity equals C_{ci}) and depend on a single parameter (the signal-to-noise ratio for the AWGN channel, the transition probability for the BSC, and the erasure probability for the BEC): given the channel capacity, this parameter is uniquely defined. The channels of the second set are binary input but asymmetric, since the transition probability ($0 \rightarrow 1$) is different from the transition probability ($1 \rightarrow 0$). In particular, for the Z channel $P\{0 \rightarrow 1\} = 0$. For the BAC two parameters are necessary. We chose to specify the ratio $t \triangleq P\{0 \rightarrow 1\}/P\{1 \rightarrow 0\}$ as a constant parameter. The MI between the input and the output of the channel, i.e., the start point of the EXIT charts, is not the channel capacity anymore, because of the asymmetry of these channels.

In Figure 3, the BER curves of all three codes, transmitted over the channels of the first set, are shown as a function of the channel capacity—note that the channel capacity can assume values between 0 and 1, since the transmitted symbols are binary. From the results in Figure 3 it is immediate to observe that the convergence threshold, in terms of capacity, depends only on the code, but not on the channel. The slight differences between the BER curves relative to the same code depend on the actual code structure, which contains cycles [4], and on the approximate nature of the

EXIT chart-based analysis technique.

In Figure 4, the BER curves of the asymmetric channels of the second set are shown. In particular for the BAC the ratio $t = 3.0$. For comparison purposes, the performance over the AWGN channel is also shown. Clearly the conjecture holds for asymmetric channels as well. Usually asymmetric channels are disregarded, because of the inherent complexity of performing analyses by means of density evolution [11].

A remark is worth while at this point. In order to get insights into the performance of LDPC codes, the right parameter to be analyzed might not be the BER, which does not explicitly appear in the EXIT charts, but rather the MI between the message set at the last iteration (at the output of variable nodes) and the transmitted sequence. Nevertheless, since the powerful LDPC codes exhibit a typical *threshold behavior*, the most interesting parameter is the threshold value. As shown, this value can be graphically evaluated in a BER-versus-capacity plot, corresponding to the cliff of the BER function.

5. Conclusions

In this paper, we have conjectured and demonstrated that good LDPC codes for a particular memoryless channel are good for any memoryless channel, in the sense that they provide similar BER in the same capacity region, regardless of the channel type. An intuitive justification has been provided and Monte Carlo simulations have been presented to support this conjecture.

A direct application of this result is to design of LDPC codes whose output binary symbols are mapped over a high-order constellation descriptive of M-ary modulations. At the receiver side, a soft demapper can generate reliability values for the mapped bits to be passed to the LDPC decoder. If no iterative exchange of information between the soft demapper and the LDPC decoder is allowed, the EXIT chart based analysis suggests that the behavior of the decoder depends only on the degree distributions of the LDPC code. The reliability values generated by the demapper can thus be re-interpreted as outputs of a memoryless channel. Hence, assuming non iterative detection, LDPC codes designed for a simple memoryless channel (e.g., BSC) will be a good choice also if mapped over high-order modulations.

References

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

- [2] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, February 1999.
- [3] S.-Y. Chung, J. G. D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, February 2001.
- [4] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, February 2001.
- [5] M. Tuechler and J. Hagenauer, "Exit charts and irregular codes," in *36th Ann. Conf. on Inf. Sci. and Syst*, Princeton, March 2002.
- [6] S. ten Brink, "Convergence of iterative decoding," *IEE Electronics Letters*, vol. 35, pp. 1117–1119, 24th June 1999.
- [7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.
- [8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, February 2001.
- [9] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727 – 1737, October 2001.
- [10] R. Urbanke, web site, "<http://lthcwww.epfl.ch/research/ldpcopt/>".
- [11] T. Richardson and R. Urbanke, "The capacity of low density parity check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, February 2001.