

OPTIMIZATION WITH EXIT FUNCTIONS OF GLDPC-STAIRCASE CODES FOR THE BEC

Ferdaouss Mattoussi*

Valentin Savin[†]

Vincent Roca*

Bessem Sayadi[‡]

*Inria, France

[†]CEA Grenoble, France

[‡]Alcatel-Lucent Bell Labs, France

{ferdaouss.mattoussi,vincent.roca}@inria.fr, {valentin.savin}@cea.fr, {bessem.sayadi}@alcatel-lucent.com

ABSTRACT

In a previous work we introduced the Generalized LDPC-Staircase codes for the Binary Erasure Channel, based on LDPC-Staircase codes and Reed Solomon as component codes. In this paper we perform an asymptotic analysis, in terms of EXtrinsic Information Transfer functions and we derive an upper bound of the ML decoding threshold based on the area theorem. We use this analysis to study the impact of the internal LDPC-Staircase code rate on the performance, and show that the proposed Generalized LDPC-Staircase codes closely approach the channel capacity, with only a small number ($E = 2, 3$) of extra-repair symbols per check node.

1. INTRODUCTION

Generalized LDPC (GLDPC) codes [1], invented by Tanner, are an extension of LDPC codes where the Single Parity Check (SPC) nodes are replaced with linear block codes, referred to as *component codes*. An advantage of GLDPC codes is that more powerful component codes can be employed (instead of SPC codes), therefore yielding better erasure recovery performance. Many researches have been carried out in order to improve GLDPC performance [2][3] [4][5][6]. The EXtrinsic Information Transfer (EXIT) functions method, introduced by Ten Brink [7][8], turned out to be a powerful tool. This method first appeared as a handy way to visualize the Iterative (IT) decoding process of LDPC codes, using EXIT curves, to easily identify the bottlenecks in the decoding process. Then the method has been used to design codes with improved performance and has been applied to GLDPC codes in [8] and in [6].

Our work relies on the GLDPC-Staircase codes [9], a class of small rate GLDPC codes (i.e. that can efficiently produce a large number of repair symbols, on demand) with interesting erasure recovery performance under IT decoding. This performance is achieved thanks to the use of the Density Evolution (DE) method [10] that enabled an optimization of the distribution of repair symbols produced by the component codes. These codes are the core of the present work, but the decoding scheme is extended: to the initial IT plus Reed-Solomon

(RS) scheme of [9], we add a Maximum Likelihood (ML) decoding scheme to further improve the code performance (the reasons why this is a good practical solution are detailed in Section 2.2). This extension, called hybrid decoding, results in exceptional performance gains (see Section 2.3).

The **contributions of our work** are twofold. First of all we derive DE equations and EXIT functions for the proposed GLDPC-Staircase codes. We follow the ideas from [11][12], where a relationship between IT and ML decoding of LDPC codes over the BEC is established, based EXIT functions and the area theorem. In our work we extend this approach to our GLDPC-Staircase codes with a hybrid decoding.

Secondly we use the EXIT method to tune an internal parameter of our codes. More precisely we explain (Section 2) that a given GLDPC-Staircase code rate (i.e. as seen by the user) can be achieved in different ways, for instance by adding a large number E of extra-repair symbols produced by RS encoding and a small number of repair symbols produced by LDPC-Staircase encoding, or vice versa. Therefore we apply the EXIT method and show that increasing the number E up to a value of 2 or 3 enables to approach the channel capacity very closely.

This paper is organized as follows. We detail GLDPC-Staircase codes in Section 2. Then we propose the DE equations of GLDPC-Staircase codes in Section 3. We extend the ideas of [13] to our case study and give the EXIT functions of the (IT+RS) and ML decoding in section 4. We apply this method and study the impacts of the internal parameter E in Section 5. Finally we conclude.

2. INTRODUCTION TO GLDPC-STAIRCASE CODES

We first introduce the GLDPC-Staircase code design, their hybrid decoding, and give a few results showing their excellent performance and flexibility.

2.1. GLDPC-Staircase code construction

GLDPC-Staircase (N_G, K) codes [9] [14] can be represented by a Tanner graph (Fig. 1) with the following meaning:

- each **check node** corresponds to a RS code based on Hankel matrices, a specific construction of RS codes

This work is supported by the ANR-09-VERS-019-02 grant (ARSSO project) and by the Inria - Alcatel Lucent Bell Labs joint laboratory.

that has the interesting property that the first repair symbol is **also** equal to the XOR sum of the source symbols. This symbol can therefore be encoded either by means of an LDPC-Staircase encoding (faster) or RS encoding. This property does not hold for the other repair symbols, called extra repair symbols;

- the **variable nodes** are broken into three categories: (1) the source symbols; (2) the first repair symbol generated by each RS code (or by the LDPC-Staircase code), that only depend on source and repair symbols (i.e. each repair symbol depends on the previous repair symbol because of the staircase structure of the LDPC code); and (3) the extra-repair symbols generated by RS codes.

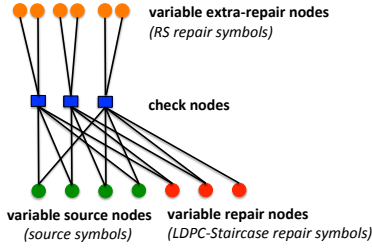


Fig. 1. GLDPC-Staircase (13, 4) code with $E = 2$.

For the reasons detailed in [14] (i.e. improved performance under ML decoding compared to an irregular distribution), we assume that all the check nodes have the same number, E , of extra-repair symbols. So for a fixed GLDPC code rate, r_G , the code rate of the LDPC-Staircase base code is given by:

$$r_L = \frac{r_G(1 + E)}{1 + r_GE} \quad (1)$$

Let N_L and K be the LDPC-Staircase code length and dimension, and N_G be the length of the GLDPC-Staircase code (which is also of dimension K). Then $N_G = N_L + n_{\text{extra}}$, where n_{extra} is the total number of extra-repair RS symbols.

Let H_L be the binary parity-check matrix of the LDPC-Staircase code, of size $M_L = N_L - K$ rows and N_L columns. H_L has the form $(H_1|H_2)$. H_1 is the $M_L \times K$ left-hand side part (information part) and each column is of degree N_1 (number of "1s" per column). H_2 is the $M_L \times M_L$ right-hand side part (redundancy part) and features a staircase (i.e. double diagonal) structure. H_1 is created in a fully regular way, in order to have constant column and row degrees. More precisely, each column of H_1 is of degree N_1 (number of "1s" per column), which is an input parameter during the LDPC-Staircase code creation [15]. Each row m of H_1 is of degree $\frac{N_1}{r_L - 1}$, and because of the staircase structure of H_2 , depending on whether $m = 1$ or $m > 1$, a row m of H_L is of degree:

$$d_1 = \frac{N_1}{\frac{1}{r_L} - 1} + 1 \quad \text{and} \quad d_{m>1} = \frac{N_1}{\frac{1}{r_L} - 1} + 2 \quad (2)$$

The E extra-repair symbols associated to the m^{th} row of H_L are generated by $RS(n_m, k_m)$ encoding over $GF(2^8)$. Here n_m and k_m are respectively the RS code length and dimension, and they are related to the other parameters of row m as follows. For row $m > 1$, the various source symbols (i.e. from the user point of view) that are involved in this row plus the previous repair symbol are considered as source symbols from the RS point of view. The new LDPC-Staircase repair symbol for this row plus the E extra-repair symbols are considered as repair symbols from the RS point of view. For the first row the only difference is the fact there is no previous repair symbol (it's the beginning of the staircase). So:

$$n_m = k_m + 1 + E \quad (3)$$

with $k_m = d_m - 1$ (no matter the row).

2.2. GLDPC-Staircase decoding

One approach consists in using an ML decoding over the full system, with operations over $GF(2^8)$, which enables to achieve the best possible erasure recovery capabilities. However, it does not take into account the complexity reduction gains made possible by GLDPC-Staircase specificities. Therefore we chose a different strategy, called hybrid decoding, that consists of a joint use of four decoders:

- *IT decoder over the binary LDPC-Staircase system:* extra-repair symbols are ignored at this step. This solution features a linear complexity with sub-optimal erasure recovery capabilities;
- *RS decoder for a given check node:* this is a classic RS decoding that takes into account the three types of symbols. It has a higher complexity but is MDS;
- *Binary ML decoder over the LDPC-Staircase system:* extra-repair symbols are once again ignored at this step. If this solution features a quadratic complexity in terms of the number of XOR operations between symbols, it allows to reach the maximum correction capabilities when ignoring extra-repair symbols;
- *Non binary ML decoder:* this solution also features a quadratic complexity but operations are now significantly more complex (performed on $GF(2^8)$) than simply XORing two symbols. However it allows reaching the maximum correction capabilities of the code. It is equivalent to the ML decoding over the full system mentioned above, but the system on which it is applied is hopefully simplified by the previous three decoders.

Decoding succeeds if one or several of these decoders succeed and recover all the missing source symbols.

2.3. GLDPC-Staircase performance

As shown in [14], GLDPC-Staircase codes exhibit excellent performances under hybrid decoding over the BEC, both for

large and very small objects, and both in terms of average performance and low error floor. For instance, with a source block of size $K = 1\,000$ symbols (resp. $K = 32$ symbols), a code rate $r_G = 1/2$, an overhead of 5 symbols (resp. 3 symbols) (i.e. after receiving 1 005 symbols (resp. 35 symbols) chosen randomly in the set of N_G symbols) is sufficient to have a decoding failure probability below 10^{-4} .¹

Several additional benefits are detailed in [14], like their small rate feature (a large number of repair symbols can be produced, in an incremental way, on demand, while keeping excellent performances) and their major flexibility (the GLDPC-Staircase behavior can be tuned to look more like MDS codes or LDPC-Staircase codes). It makes it possible to adapt to the exact use-case and channel conditions (e.g. when used in fountain like applications, e.g. within a FLUTE carousel [16], over wireless networks).

3. DENSITY EVOLUTION (DE) EQUATIONS

3.1. Preliminaries

In the sequel, we denote by \hat{d}_v and \hat{d}_c the maximum variable and check node degrees in the bipartite (Tanner) graph associated with the LDPC-Staircase code. Following [10], we define the edge-perspective Degree Distribution (DD) polynomials by $\lambda(x) = \sum_{d=1}^{\hat{d}_v} \lambda_d x^{d-1}$ and $\rho(x) = \sum_{d=1}^{\hat{d}_c} \rho_d x^{d-1}$, where λ_d (resp. ρ_d) represents the fraction of edges connected to variable-nodes (resp. check-nodes) of degree d . From a node perspective, the DD polynomials are given by $L(x) = \sum_{d=1}^{\hat{d}_v} L_d x^d$ and $R(x) = \sum_{d=1}^{\hat{d}_c} R_d x^d$ where L_d (resp. R_d) represents the fraction of variable-nodes (resp. check-nodes) of degree d .

Given a GLDPC-Staircase code, DD polynomials λ and ρ are defined by the underlying LDPC-Staircase code, defined by the bottom graph of Figure 1 (that is, not containing the extra-repair nodes). We denote by $\mathcal{E}(\lambda, \rho, E)$ the ensemble of GLDPC-Staircase with edge-perspective DD polynomials λ and ρ , and with E extra-repair symbols per check-node.

Assume that an arbitrary code from $\mathcal{E}(\lambda, \rho, E)$, of length N_G , is used over the BEC, and let ε denote the channel erasure probability. The probability threshold of the ensemble $\mathcal{E}(\lambda, \rho, E)$ is defined as the supremum value of ε (that is, the worst channel condition) that allows transmission with an arbitrary small error probability, assuming that N_G goes to infinity. The threshold value of a given ensemble of codes can be efficiently computed by using the *Density Evolution* (DE) method [10], which recursively computes the fraction of erased messages passed during the belief propagation decoding. Density evolution equations are derived in the next section, by using the methodology introduced in [9].

¹In these tests the LDPC-Staircase code rate is set to $r_L = 2/3$ and $N_1 = 5$. Clearly, with $K = 32$ symbols, an option is to set $r_L = 1$ so that only extra-repair symbols are used, which means that GLDPC-Staircase codes are turned into MDS codes.

3.2. DE equations for GLDPC-Staircase codes

In this section we derive the DE equation for the ensemble $\mathcal{E}(\lambda, \rho, E)$ of GLDPC-Staircase codes. In the sequel, the degree of a check or variable (source or repair) node will always refer to its degree in the underlying LDPC-Staircase code. We are interested in the erasure probability of messages exchanges by the IT+RS decoding along the messages of the LDPC-Staircase code. We denote by P_ℓ , the probability of a LDPC symbol (source or repair) node sending an erasure at iteration ℓ . Similarly, Q_ℓ denotes the probability of a check node sending an erasure (to an LDPC symbol-node) at iteration ℓ . Clearly, P_0 is equal to the channel erasure probability ε . In order to derive a recursive relation between P_ℓ and $P_{\ell+1}$, we proceed as follows.

Consider a constraint node c connected to symbol-nodes $(v_1, \dots, v_d, e_1, \dots, e_E)$ where v_i denotes an LDPC (source or repair) symbol node and e_i denotes an extra-repair node. Since c corresponds to an RS code, it can recover the value of an LDPC symbol node, say v_1 , if and only if the number of erasures among the other symbol-nodes (v_2, \dots, v_d) is less than or equal to E . Now, at iteration ℓ , the LDPC symbols are erased with probability P_ℓ , while extra repair symbols are always erased with probability ε , the channel erasure probability. It follows that the probability of a check node of degree d recovering the value of an LDPC symbol at iteration $\ell + 1$, denoted by $\bar{Q}_{\ell+1}(d)$, is given by:

$$\bar{Q}_{\ell+1}(d) = \sum_{\substack{0 \leq i < d, 0 \leq j \leq E \\ i+j \leq E}} \binom{d-1}{i} P_\ell^i (1-P_\ell)^{d-1-i} \binom{E}{j} \varepsilon^j (1-\varepsilon)^{E-j} \quad (4)$$

Averaging over all possible values of d , we get:

$$Q_{\ell+1} = 1 - \sum_{d=1}^{\hat{d}_c} \rho_d \bar{Q}_{\ell+1}(d) \quad (5)$$

Conversely, an LDPC symbol node v of degree d , connected to check nodes c_1, \dots, c_d , sends an erasure to the check node c_1 iff it was erased by the channel, and it received erased messages from all check nodes c_2, \dots, c_d . Since this happens with probability $\varepsilon \cdot Q_{\ell+1}^{d-1}$, and averaging over all possible degrees d , we get:

$$P_{\ell+1} = \varepsilon \sum_{d=1}^{\hat{d}_v} \lambda_d Q_{\ell+1}^{d-1} = \varepsilon \lambda(Q_{\ell+1}) \quad (6)$$

Using equations (4), (5), (6) we can determine a recursive relation between P_ℓ and $P_{\ell+1}$, with $P_0 = \varepsilon$. The decoder can recover from a fraction of ε erased symbols iff $\lim_{\ell \rightarrow +\infty} P_\ell = 0$. Therefore, the threshold probability can be computed by:

$$\varepsilon^{(\text{IT+RS})}(\lambda, \rho, E) = \max\{P_0 \mid \lim_{\ell \rightarrow +\infty} P_\ell = 0\} \quad (7)$$

When no confusion is possible, the above threshold value will be simply denoted by $\varepsilon^{(\text{IT+RS})}$.

4. EXIT FUNCTIONS AND ML THRESHOLD UPPER BOUND

EXtrinsic Information Transfer (EXIT) curves were first introduced in [7] as a technique to analyse the convergence of iterative decoding process of parallel concatenated component codes. For binary LDPC codes, a slightly different definition of the EXIT curve has been introduced in [11], where the EXIT curve is associated with the sparse graph *system* rather than with *component* codes). Roughly speaking, the EXIT curve gives the fraction of erased bits “contained” in the *extrinsic* information produced by the decoding algorithm, assuming that the code length tends to infinity. The EXIT curve can be defined for any decoding algorithm (e.g. IT or ML decoding), and it relates to the asymptotical performance of an ensemble of codes under the considered decoding. Obviously, in case of IT decoding, there a tight relation between the EXIT curve and the density evolution equations derived in the previous section. This relation will be discussed for GLDPC-Staircase codes under (IT+RS) decoding in Section 4.1. For the ML decoding, it has been shown in [11] that the area under the EXIT curve is always equal to the asymptotic rate of the ensemble. This allows deriving an upper bound of the ML threshold, which is conjectured to be tight in a quite general settings, especially for codes defined by almost regular graphs. We extend this technique to our GLDPC-Staircase codes in Section 4.2

4.1. EXIT curve for the (IT+RS) decoding

The EXIT curve defined in this section relates to the asymptotical performance of the ensemble $\mathcal{E}(\lambda, \rho, E)$ under the (IT+RS) decoding. Precisely, we start with a fixed number of decoding iterations, say ℓ , and let the code length tend to infinity. Within this limit, we denote by $p(\varepsilon)$ the probability of the *extrinsic* information² of a random LDPC symbol-node being erased. Using the DE equations from Section 3.2, we obtain $p(\varepsilon) = \sum_{d=1}^{\hat{d}_v} L_d Q_\ell^d = L(Q_\ell)$, where L is the node-perspective DD polynomial of LDPC symbol-nodes. The EXIT curve of the (IT+RS) decoding is defined as the limit of the above probability when the number of iterations goes to infinity, that is, $h^{(\text{IT+RS})}(\varepsilon) = \lim_{\ell \rightarrow +\infty} L(Q_\ell)$. Since L is a finite-degree polynomial, we can also write:

$$h^{(\text{IT+RS})}(\varepsilon) = L(Q_{+\infty}), \quad (8)$$

where $Q_{+\infty} = \lim_{\ell \rightarrow +\infty} Q_\ell$. As a consequence, the probability threshold and the EXIT-curve satisfy the following equality:

$$\epsilon^{(\text{IT+RS})} = \sup\{\varepsilon \in [0, 1] \mid h^{(\text{IT+RS})}(\varepsilon) = 0\} \quad (9)$$

An example of EXIT function under the (IT+RS) decoding, for an ensemble of GLDPC codes, will be discussed in the next section.

²The information we get by tacking into account the messages from the neighbor check-nodes, but not the channel output

4.2. EXIT curve for the ML decoding

As for the (IT+RS) decoding, the EXIT curve of the ML decoding is also defined in terms of *extrinsic erasure probability*. Precisely, in the limit of infinite code length, for a given channel erasure probability ε , $h^{\text{ML}}(\varepsilon)$ is the probability of a symbol node being erased after ML decoding, assuming that the received value (if any) of this particular symbol has not been submitted to the decoder. The ML probability threshold is given by:

$$\epsilon^{\text{ML}} = \sup\{\varepsilon \in [0, 1] \mid h^{\text{ML}}(\varepsilon) = 0\} \quad (10)$$

The exact computation of the EXIT function for the ML decoding is a difficult task. However, using the area theorem³ [11], we get $\int_{\epsilon^{\text{ML}}}^1 h^{\text{ML}}(\varepsilon) d\varepsilon = r_G$, where r_G is the designed coding rate of the given ensemble of GLDPC codes. Moreover, since the (IT+RS) decoding is suboptimal with respect to the ML decoding, we have $h^{(\text{IT+RS})}(\varepsilon) \geq h^{\text{ML}}(\varepsilon)$. Hence, if for some $\bar{\epsilon}^{\text{ML}}$

$$\int_{\bar{\epsilon}^{\text{ML}}}^1 h^{(\text{IT+RS})}(\varepsilon) d\varepsilon = r_G, \quad (11)$$

we necessarily have $\bar{\epsilon}^{\text{ML}} \geq \epsilon^{\text{ML}}$. This gives an upper bound on the ML-threshold, which is known to be tight in the binary case.

Fig. 2 shows the EXIT curve, under (IT+RS) decoding, for the ensemble of GLDPC codes with node-perspective DD polynomials $L(x) = 0.4x^2 + 0.6x^5$ and $R(x) = x^{10}$, and with $E = 2$ extra repair symbols per check-node. We note that the LDPC code is of rate $r_L = 3/5$, with all source symbol-nodes of degree 5 and all repair symbol-nodes of degree 2. The coding rate of the GLDPC code is given by $r_G = 1/3$. The (IT+RS) threshold value is $\epsilon^{(\text{IT+RS})} = 0.5376$. It can be seen that $h^{(\text{IT+RS})}(\varepsilon) = 0$ for values $\varepsilon < \epsilon^{(\text{IT+RS})}$, then for $\varepsilon = \epsilon^{(\text{IT+RS})}$ it jumps to a non-zero value and increases until it reaches a value of 1 for $\varepsilon = 1$. The ML-threshold upper-bound is the unique point $\bar{\epsilon}^{\text{ML}} \in [\epsilon^{(\text{IT+RS})}, 1]$ such that the hatched area below the (IT+RS)-EXIT curve, delimited by $\varepsilon = \bar{\epsilon}^{\text{ML}}$ at the left and by $\varepsilon = 1$ at the right, is equal to the GLDPC rate $r_G = 1/3$. In this case, we obtain $\bar{\epsilon}^{\text{ML}} = 0.6664$.

5. APPLICATION: CHOOSING A VALUE FOR E

Let us consider a GLDPC-Staircase code of rate r_G . Several values of E , or equivalently of the internal LDPC-Staircase code rate r_L , enable to achieve this global code rate (see Eq. 1). However choosing a value impacts the performance achieved. Therefore we now apply the techniques developed in section 4 to adjust E , by computing the upper bound on the ML threshold for several values of E . These results are

³Note that the area theorem applies for the MAP decoding, but over the BEC, MAP and ML decoding are equivalent.

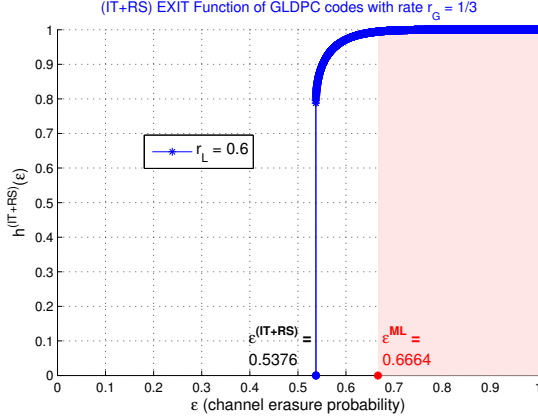


Fig. 2. Example of EXIT function, (IT+RS) threshold value, and ML threshold upper bound for an ensemble of GLDPC codes

Table 1. $\bar{\epsilon}_{ML}$ of GLDPC codes as a function of r_G

r_G	$E=0$	$E=1$	$E=2$	$E=3$	$E=4$	$E=5$	δ_{sh}
1/3.5	0.7054	0.7124	0.7138	0.7141	0.7142	0.7142	0.7142
1/3	0.6634	0.6652	0.6664	0.6665	0.6666	0.6666	0.6667
1/2	0.4946	0.4993	0.4999	0.4999	0.4999	0.4999	0.5000
2/3	0.3301	0.3330	0.3333	0.3333	0.3333	0.3333	0.3333
3/4	0.2484	0.2498	0.2499	0.2499	0.2499	0.2499	0.2500
9/10	0.0991	0.0999	0.0999	0.0999	0.0999	0.0999	0.1000

summarized in Table 1 and compared to the capacity limit (δ_{sh}). We notice that increasing E (or equivalently increasing the LDPC code rate) quickly increases the upper bound on the ML threshold, until it reaches a stable value very close to the Shannon limit δ_{sh} . Depending on r_G this stable value is obtained with $E = 1, 2$ or 3 . Therefore, a small number of extra-repair RS-symbols per check-node is sufficient to get extremely close to the channel capacity.

Let us now compare these theoretical results with practical results, obtained by simulations with a software GLDPC-Staircase codec that we designed. We see in Fig. 3 that the average erasure recovery performance (measured here in terms of inefficiency, i.e. the ratio between the number of symbols needed for decoding to succeed and k) quickly approaches 1 (i.e. no overhead, decoding is possible with exactly k symbols) as $E = 3$, even for very small code dimension.

6. CONCLUSIONS

In this paper we have analyzed the asymptotic behavior of GLDPC-Staircase codes for the BEC, through the DE and EXIT functions methods. More specifically, after explaining how these techniques apply to our use-case, we used them to tune an internal parameter of the code. Our results indicate that increasing the number E of extra repair symbols quickly increases the upper bound on the ML thresholds until it reaches a stable value very close to the Shannon limit. In practice choosing $E = 2$ or 3 yield performance that approach those of an ideal, MDS code, even for very small code

dimensions.

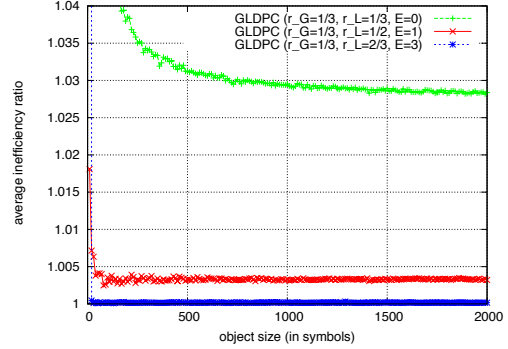


Fig. 3. Finite length analysis of the impacts of E .

7. REFERENCES

- [1] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [2] G. Yue, L. Ping, and X. Wang, "Generalized low-density parity-check codes based on Hadamard constraints," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 1058–1079, 2007.
- [3] M. Lentmaier and K.S. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Communications Letters*, vol. 3, no. 8, pp. 248–250, 1999.
- [4] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *IEEE International Conference on Communications*, 1999, vol. 1, pp. 441–445.
- [5] Y. Wang and M. Fossorier, "Exit chart analysis for doubly generalized ldpc codes," in *Proc. IEEE Global Telecommun. Conf.*, 2006, pp. 1–6.
- [6] E. Paolini, M. Fossorier, and M. Chiani, "Analysis of doubly-generalized LDPC codes with random component codes for the binary erasure channel," in *Proceedings of Allerton Conference on Communications, Control and Computing*, 2006.
- [7] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [8] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, 2004.
- [9] M. Cunche, V. Savin, V. Roca, G. Kraidy, A. Soro, and J. Lacan, "Low-rate coding using incremental redundancy for GLDPC codes," in *IEEE Int. Workshop on Satellite and Space Comm.*, 2008, pp. 299–303.
- [10] T.J. Richardson and R.L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [11] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell's construction: The hidden bridge between maximum-likelihood and iterative decoding," in *IEEE Int. Symp. on Information Theory*, 2004, p. 225.
- [12] C. Méasson, *Conservation laws for coding*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2006.
- [13] C. Méasson and R. Urbanke, "An upper-bound for the ML threshold of iterative coding systems over the BEC," in *Proc. of the 41st Allerton Conf. on Communications, Control and Computing*, October 2003, p. 3.
- [14] F. Mattoussi, V. Roca, and B. Sayadi, "Design of small rate, close to ideal, GLDPC-Staircase codes for the erasure channel," 2012, submitted for publication.
- [15] V. Roca, C. Neumann, and D. Furodet, "Low density parity check (LDPC)-staircase and triangle forward error correction (FEC) schemes," June 2008, IETF RMT Working Group, RFC 5170 (Proposed Standard).
- [16] R.T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh, "FLUTE - File Delivery over Unidirectional Transport," October 2004, IETF RMT Working Group, Request For Comments RFC 3926.