ISIT'2013 1569719691

Improving the BP estimate over the AWGN channel using Tree-structured Expectation Propagation

Luis Salamanca, Juan José Murillo-Fuentes
Department of Signal Theory and Communications
University of Sevilla.
email:{salamanca,murillo}@us.es

Pablo M. Olmos, Fernando Pérez-Cruz
Department of Signal Theory and Communications
University Carlos III in Madrid.
email: {olmos,fernando}@tsc.uc3m.es

***Abstract*—In this paper, we propose the tree-structured expectation propagation (TEP) algorithm for low-density parity-check (LDPC) decoding over the binary additive white Gaussian noise (BI-AWGN) channel. By approximating the posterior distribution by a tree-structure factorization, the TEP has been proven to improve belief propagation (BP) decoding over the binary erasure channel (BEC). We show for the AWGN channel how the TEP decoder is also able to capture additional information disregarded by the BP solution, which leads to a noticeable reduction of the error rate for finite-length codes. We show that for the range of codes of interest, the TEP gain is obtained with a slight increase in complexity over that of the BP algorithm. An efficient way of constructing the tree-like structure is also described.**

***Index Terms*—Channel Coding, LDPC codes, Expectation Propagation**

I. Introduction

Although Gallager proposed low-density parity-check (LDPC) codes along with linear time practical algorithms [1], its rebirth came with the popularization of belief propagation (BP) as an efficient message-passing algorithm to estimate the bit marginals from the posterior distribution [2], [3]. While exact for tree-like graphs, the BP solution is suboptimal for LDPC codes of interest [4]. In the last decade, the BP estimate has been improved by designing sparse codes that exploit the algorithm behavior, for instance irregular LDPC codes [5] or convolutional LDPC codes [6], [7]. Nonetheless, little work has been done in the coding community to explore alternative inference techniques that improve the BP estimate. In the machine learning literature, there are several extensions to improve BP performance in loopy graphs [8]–[10], but they are typically thought for dense graphs, where BP is impractical. Indeed, one of the few practical cases where one of these methods has been successfully applied to LDPC decoding is generalized BP (GBP) over discrete channels with memory [11], in which the channel structure favors the cycles in the graph and GBP is able to decode accurately.

Expectation propagation (EP) [12] is an inference method to construct tractable approximations $q(\mathbf{V})$ of a joint pdf $p(\mathbf{V})$. In [9], [12], it is shown that the BP update rules are equivalent to EP when we consider fully factorized approximations, i.e. $q(\mathbf{V}) = q_1(V_1)q_2(V_2)\dots q_{\mathtt{n}}(V_{\mathtt{n}})$. In a novel work, we proposed the TEP decoder for LDPC codes over the BEC [13], borrowing from the EP algorithm using a tree-structure approximation [10]. For the BEC case, the decoder performance has been analyzed in detail. Complexity issues, asymptotic behavior and finite-length scaling laws were studied in [13]. It is shown that the TEP decoder has an asymptotic performance identical to the BP decoder, but that it is able to outperform BP for finite-length codes. This proposal was also extended to the maximum likelihood solution in [14].

For the BEC, there are several proposals [4], [15], [16] that improve BP performance over the BEC with varying degree of success and complexity level. All of them rely on the simple structure of the BEC to achieve those gains and cannot be applied to general binary memoryless symmetric (BMS) channels. In this paper, we show that the TEP decoder can be used for LDPC decoding over any channel model and we specifically focus on the binary additive white Gaussian noise (BI-AWGN) channel. For a given tree-structure, the TEP procedure iteratively finds the pairwise factors that minimize the KL divergence with the true posterior [12]. As we show, computing local pairwise marginals at each check node to update the tree approximation $q(\mathbf{V})$ yields into a more accurate inference procedure than the BP update rules, able to deal with short and medium-sized loops in the LDPC code. As a consequence, we observe an improved performance in both the waterfall and the error floor regions. The TEP complexity is greatly minimized by following some simple rules presented in the paper, resulting in a slight increase over the BP complexity.

II. Tree-structured expectation propagation

Given a parity check matrix $\mathbf{H}$, of dimensions $\mathtt{k}\times\mathtt{n}$, which defines a binary LDPC code of rate $\mathtt{r} = 1-\mathtt{k}/\mathtt{n}$, a codeword $\mathbf{v}$ belongs to the code if $\mathbf{v}\mathbf{H}^\top = \mathbf{0}$. Assuming equiprobable codewords $\mathbf{V}$, the posterior distribution factorizes as

$$p(\mathbf{V}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{V}) \prod_{j=1}^{\mathtt{k}} \mathtt{C}_j(\mathbf{V}), \tag{1}$$

where $\mathtt{C}_j(\mathbf{V})$ is the parity constraint described by the j-th row of $\mathbf{H}$ and $p(\mathbf{y}|\mathbf{V})$ is the channel likelihood. Although EP can be applied to any channel model, we focus on a memoryless channel, where $p(\mathbf{y}|\mathbf{V}) = \prod_{i=1}^{\mathtt{n}} p(y_i|V_i)$.

Minka and Qi in [10] proposed the EP algorithm together with a tree-structure approximation, hereafter referred as TEP

(Tree EP), to approximate $p(\mathbf{V}|\mathbf{y})$:

$$p(\mathbf{V}|\mathbf{y}) \approx q(\mathbf{V}) = \prod_{i=1}^{\mathtt{n}} q_i(V_i|V_{p_i}), \tag{2}$$

where these joint factors relate pairs of variables (i, p_i) and approximate the true conditional probabilities $p(V_i|V_{p_i}, \mathbf{y})$. Since this joint information is captured and refined during the whole decoding process, EP typically obtains better estimates for all marginals $p(V_i|\mathbf{y})$ than BP [9], [10]. As we show in the paper, including a cycle free set of pairwise marginals helps to reduce the negative effects of loops in the graph. Notice that $p_i = \emptyset \ \forall i$ leads to a fully disconnected approximation, case for which EP is equivalent to BP because they both are just iterative methods to compute stationary points of the same free energy approximation [9], [12].

For a fixed choice of the pairs (V_i, V_{p_i}) $i = 1\ldots, \mathtt{n}$, the optimal solution for $q(\mathbf{V})$ is obtained by minimizing the inclusive Kullback-Leibler (KL) divergence:

$$\widehat{q}(\mathbf{V}) = \arg \min_{q(\mathbf{V}) \in \mathcal{F}_{tree}} D_{KL}\Big(p\left(\mathbf{V}|\mathbf{y}\right)||q(\mathbf{V})\Big), \tag{3}$$

where $\mathcal{F}_{tree}$ is the family of all discrete probability distributions that factorize according to (2). The solution to the problem in (3) is given by moment matching [9]: $\widehat{q}(\mathbf{V})$ is such that it has the same moments (pairwise marginals) than $p(\mathbf{V}|\mathbf{y})$ [12]. However, computing exact marginals of $p(\mathbf{V}|\mathbf{y})$ is unfeasible since it is equivalent to do optimal decoding.

To efficiently find a solution to the problem (3), Minka [12] proposed a sequential algorithm to iteratively approach $\widehat{q}(\mathbf{V})$, where only one parity-check function is considered at each iteration. For the LDPC decoding case, the algorithm in detail can be found in [13], we just describe here the main steps. First, we construct $q(\mathbf{V})$ in $\mathcal{F}_{tree}$ by substituting each parity factor $\mathtt{C}_j(\mathbf{V})$ in (1) with a function $W_j(\mathbf{V})$ that shares the factorization in (2):

$$\mathtt{C}_j(\mathbf{V}) \approx W_j(\mathbf{V}) = \prod_{z=1}^{\mathtt{n}} w_{z,j}(V_z, V_{p_z}), \tag{4}$$

where $w_{z,j}(V_z, V_{p_z})$ are non-negative real functions. At iteration ℓ, the TEP chooses a factor to refine, $m \in [1, \mathtt{k}]$, and computes a new estimate of $\widehat{q}(\mathbf{V})$, i.e $q^\ell(\mathbf{V})$, as follows:

A) Compute the auxiliary function $f(\mathbf{V}, \ell, m)$ to refine the m-th factor $W_m(\mathbf{V})$:

$$f(\mathbf{V}, \ell, m) = \mathtt{C}_m(\mathbf{V})\frac{q^{\ell-1}(\mathbf{V})}{W_m(\mathbf{V})}. \tag{5}$$

B) Estimate $q^\ell(\mathbf{V})$ by minimizing the KL divergence, done by moment matching as well,

$$q^\ell(\mathbf{V}) = \arg \min_{q(\mathbf{V}) \in \mathcal{F}_{tree}} D_{KL}\Big(f(\mathbf{V}, \ell, m)||q(\mathbf{V})\Big). \tag{6}$$

C) Update the factor $W_m(\mathbf{V})$ according to the new estimated function $q^\ell(\mathbf{V})$.

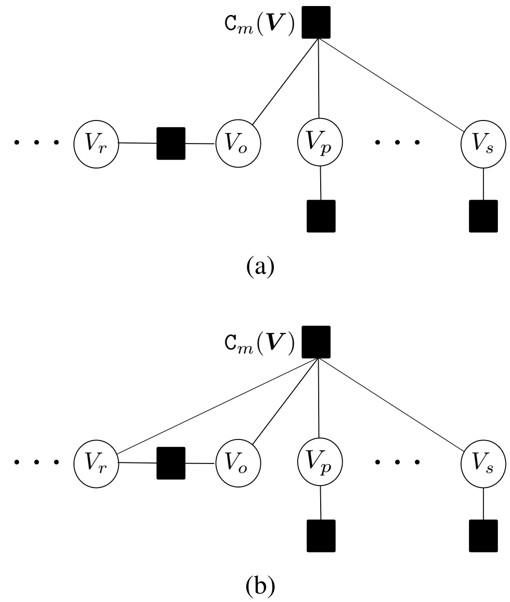


Fig. 1: Examples of factor graphs associated to $f(\mathbf{V}, \ell, m)$. In (a), the graph is cycle free since $\mathtt{C}_m$ does not connect any variable V_i with its parent V_{p_i}. In (b), the graph has a cycle since both V_r and $V_{p_r} = V_o$ belong to I_m.

Matching the moments in Step B corresponds to estimate the marginals, $q_i^\ell(V_i)$, and pairwise marginals, $q_i^\ell(V_i, V_{p_i})$, of those variables that form function $f(\mathbf{V}, \ell, m)$. This leads to two possible scenarios, as depicted in Fig. 1. In the first scenario, Fig. 1(a), none of the pairs of variables that form the tree are connected to $\mathtt{C}_m(\mathbf{V})$. The marginals can be computed by message passing at linear cost since the graph is cycle-free. This is the case for a completely disconnected structure, equivalent to BP as discussed before. For the sake of the complexity comparison between the TEP and BP, we assume that all BP steps correspond to the scenario in Fig. 1(a).

In the second scenario, Fig. 1(b), the linked variables form a cycle. The TEP algorithm computes the joint marginals using the Pearl's cutset conditioning algorithm [17]. It computes the marginals by breaking the cycles assuming a set of variables as known. The complexity is then exponential with the number of assumed variables, but the calculations are performed by message passing. For practical LDPC codes the maximum check node degree is bounded and so it is the maximum number of assumed variables. Indeed, as we show in Section V-A, we typically face only one loop and hence the complexity of dealing with this scenario is just twice that of the disconnected case.

III. Tree-structure construction

Prior to running the TEP algorithm, it is necessary to fix a tree-structure, i.e. choose a family $\mathcal{F}_{tree}$. As discussed in [12], while determining the cost of a given choice is straightforward, estimating the accuracy for each case is a non-trivial problem. In [10], it is suggested to link the variables with highest mutual information. However, for LDPC codes each pair of variables

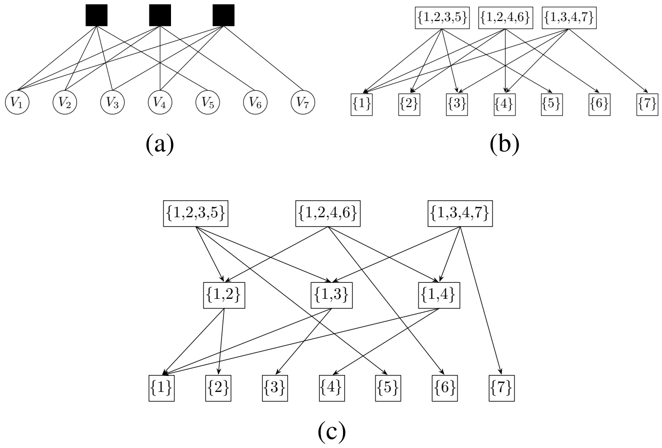


Fig. 2: Factor graph of an LDPC code (a), equivalent region graph (b) and region graph obtained through cluster variation method (c).

has a priori zero mutual information, and we thus need to find an alternative way for building the tree-structure.

For the BEC in [13], we define the tree-structure for each received word. We first run the BP algorithm until it gets stuck. We then link a pair of erased variables that are connected to a factor that has only two erased variables. This follows the highest mutual information suggestion in [10], because if there are only two erased variables in a factor node, it means that those variables are either equal or opposite. The TEP decoder over the BEC keeps on building the graph, i.e. joining pair of variables in factors with only two erased variables, until it forms a loop and we are then able to decode a set of variables that could not be decoded by BP otherwise.

For other BMS channels, the optimal method to construct the tree-structure remains open. The following simple rules work remarkably well in practice, as they emulate how the TEP decoder proceeds over the BEC. First, we run BP for an enough number of iterations. If BP does not converge to a valid codeword, we take those variables whose log-likelihood ratio (llr) oscillates around zero and those variables connected to incorrect check nodes, i.e. $\mathbf{v}\mathbf{h}_i^\top \neq 0$ where $\mathbf{h}_i$ is the i-th row of $\mathbf{H}$. This set of chosen variables is used to build the tree-structure as follows[1]:

A) Select a variable with an llr close to zero. This variable is used as a seed to construct the tree.
B) Place a pairwise factor between the latest variable included in the tree and one of its neighbors, the one that presents the llr closest to zero.
C) Repeat Step B until we go back to the seed of the tree. This last pairwise factor is not included to avoid creating a loop between the pairwise terms.
D) Go back to Step A to compute other trees, excluding the variables already selected.

Note that the above procedure does not do an exhaustive search, since we just follow neighbors until we close the loop. Hence, constructing the chain has a complexity linear in the number of variables. For signal-to-noise ratios (SNRs) corresponding to the BP waterfall region, we tend to construct a few trees containing a large set of variables since the cycles that dominate the BP performance have lengths $\mathcal{O}(\mathtt{n})$ [4]. In the error floor region, we construct several trees with few variables. Thus, for all ranges of SNR, the TEP algorithm captures valuable information disregard by the BP algorithm.

[1]The rest of variables are left unlinked in the approximation.

IV. GBP ALGORITHM FOR LDPC DECODING

In [8], Yedidia *et al.* proved that the BP message passing update rules define an iterative Lagrangian method to solve the Bethe variational problem. The Kikuchi method and related approximations improve the accuracy of the Bethe approximation by including the effect of larger cluster of variables [9]. Generalized belief propagation (GBP) is a natural generalization of the ordinary BP update rules to find fixed-points of Kikuchi-type approximations [9]. Region graphs (RGs) are the natural graphical representation for the GBP algorithm [8]. In region graphs, nodes embed information of a unique or a set or random variables. For instance, in Fig. 2(b) we include the simplest region graph associated to the LDPC factor graph in Fig. 2(a). GBP applied over this region graph provides the BP solution [18]. To improve the BP solution, it is necessary to extend the region graph, including new region nodes obtained from the intersection of larger regions. This process is known as the cluster variation method [8]. For example, in Fig. 2(b) we can add 3 new regions ($\{1,2\}$, $\{1,3\}$ and $\{1,4\}$) by intersecting the 3 larger sets. When run over the region graph in Fig. 2(c), GBP provides more accurate estimates of the bitwise marginal probabilities.

One drawback of the GBP algorithm is the complexity of the cluster variation method, i.e. the search for intersections between regions. This complexity increases with the number of variables involved and the sparsity of the graph. Thus, LDPC codes are a hard problem for the cluster variation method. However, the main problem of GBP for LDPC decoding is clearly stated in [8] by Yedidia *et al.*: *since the intersections in the region-graph correspond to loops of size four in the graph, in good LDPC codes such intersections do not exist, and consequently GBP can not improve the BP solution.* This explains why GBP has not been applied to LDPC decoding, except for transmission over channels with memory [11], [18], where the intersections are favored by the channel memory.

The TEP decoder overcomes both problems. The method proposed in Section III establishes pairwise relations without an exhaustive search in the whole graph, as the algorithm only introduces pairwise relations where BP fails. Also, since the TEP decoder imposes trees of pairwise relations, it considers and propagates mutual information in loops with sizes larger than four, improving the BP solution even for good LDPC codes, as we illustrate in the next section.

V. SIMULATION RESULTS

In this section, we illustrate the performance of the TEP decoder for different LDPC ensembles and some experimental measurements regarding its complexity with respect to the BP algorithm. We consider regular and irregular LDPC codes of moderate code-length, between a few hundred to a thousand bits, where the presence of cycles degrades the BP performance. This is the scenario in which the TEP presents the highest applicability, as random ensembles might not provide good performances for short codes and, for longer codes, BP is already close to asymptotic performance [4]. The presented results show the TEP decoder as an attractive tool to enhace the BP performance for sparse codes of moderate length.

We restrict our simulations to codes for which the minimum stopping set (SS) contains no less than $s_{\min} = s_{\min}(\mathtt{n})$ bits, where $s_{\min}(\mathtt{n})$ is the expurgation parameter, detailed in [4]. This expurgation allows to claim that the observed gain is due to a more accurate marginal estimation for cycles of size at least $s_{\min}$ bits. In Fig. 3, we show the bit error rate (BER) for the (3,6)-regular LDPC ensemble over the AWGN channel with code-lengths $\mathtt{n} = \{128, 256, 512, 1024\}$ bits, whose respective expurgation values are $s_{\min} = \{6, 6, 12, 24\}$. The results are averaged over 100 random samples of the ensemble. For the waterfall region, the gain at $\mathrm{BER} = 10^{-4}$ varies from 0.3dB for $\mathtt{n} = 128$ to 0.2 dB for $\mathtt{n} = 1024$. For the error floor region, the gain is of almost an order of magnitude across $\mathtt{n}$. In Fig. 4, we reproduce the experiments for an irregular code whose degree distribution is given by $\lambda(x) = 1/6x + 5/6x^3$ and $\rho(x) = x^5$ [4]. For this ensemble, we use code-lengths of $\mathtt{n} = \{128, 256, 512\}$ bits and their respective expurgation values are set to $s_{\min} = \{6, 12, 18\}$.

Two main conclusions can be drawn from these results. First, in scenarios with a pronounced error floor, the TEP decoder significantly improves the BP in this region. This gain suggests that there is a large fraction of critical short loops that are now broken thanks to the inference of pairwise relationships. Second, we are able to improve the BP solution in the waterfall region, proving that the EP tree-structure improves the inference accuracy over sparse graphs with very large loops, of length $\mathcal{O}(\mathtt{n})$. In spite the improvement in the waterfall region decreases with the code-length $\mathtt{n}$, it is still observable for code-lengths around a thousand bits.

A. Evaluation of the decoding complexity

As we stated in Section II, in a code with maximum check degree $r_{\max}$, the maximum complexity per TEP iteration, over the cost of processing the cycle free scenario, is upper-bounded by $2^{r_{\max}-1}$. Conversely, we show in this section how the average cost of the TEP decoder's iteration we observe in practice is much lower.

One of the main aspects to be analyzed when comparing the TEP and BP complexity is, first of all, the fractions of iterations, hereafter referred as L, for which we have to do inference in a loopy scenario by Pearl's cutset algorithm. Secondly, let's denote the average number of variables in the loops as d. Thereby, the average Pearl's cutset algorithm cost normalized by the cost of processing the cycle free scenario results in 2^{d-1}. From these two measures, the average TEP

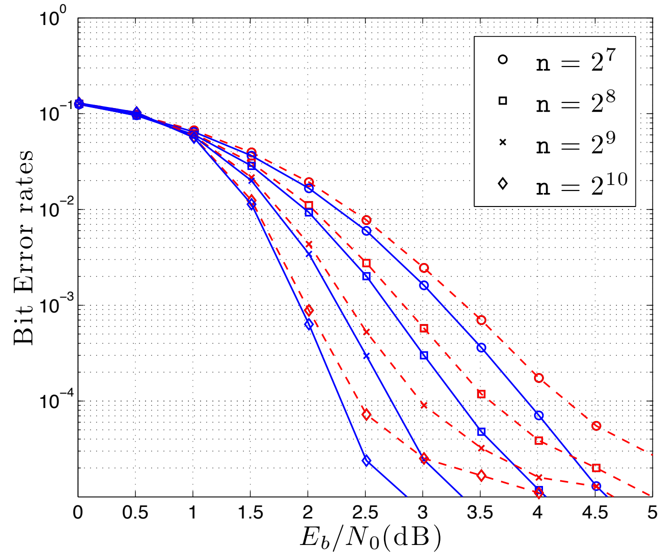


Fig. 3: BER over the AWGN channel for the BP (dashed lines) and the TEP (solid lines) decoders for the expurgated (3,6)-regular LDPC ensemble, and different code-lengths $\mathtt{n}$, $128(\circ)$, $256(\square)$, $512(\times)$ and $1024(\Diamond)$.

complexity per iteration normalized by the BP complexity is obtained as $c_{\text{TEP}} = 2^{d-1}L + (1 - L)$.

We illustrate these simple statistics computed across the experiments for the regular $(3, 6)$ code presented above. The results included are averaged over more than 10^4 received codewords. In Fig. 5(a), we depict the average cost of processing loopy scenarios through Pearl's cutset algorithm, i.e. the average value of 2^{d-1}, for $\mathtt{n} = 256$ (solid lines) and $\mathtt{n} = 512$ (dashed lines). As we observe, the typical size of the loop is $d = 2$ and, thus, the cost of processing these scenarios is only two times the cost of a BP iteration. In Fig. 5(b), we plot $(1 - L)$, namely the fraction of iterations for which $f(\mathbf{V}, \ell, m)$ is cycle free. The values are closed to 1 and tend to it with the SNR. Finally, in Fig. 5(c), we illustrate the average cost of the TEP iteration, obtained from the previous results. As we can observe, although the cost of processing the loopy scenario is at least two times more complex, the number of occurrences of this scenario is quite small and thereby the average complexity per iteration is almost that of the BP algorithm. Thus, the simulations conclude that the gain in performance is remarkable for the finite-length regime at the cost of an slight increase over the BP complexity.

VI. Conclusions

In this paper, we successfully apply the TEP decoder to the AWGN channel, which requires to generalize the formulation of the TEP decoder that in our previous works was benefited by the simple structured of the BEC. The results demonstrates that, thanks to the tree/forest structure of pairwise marginals, the TEP decoder improves the BP solution to converge to the transmitted codeword. This procedure reduces both the error rate in the waterfall and error floor regions by building

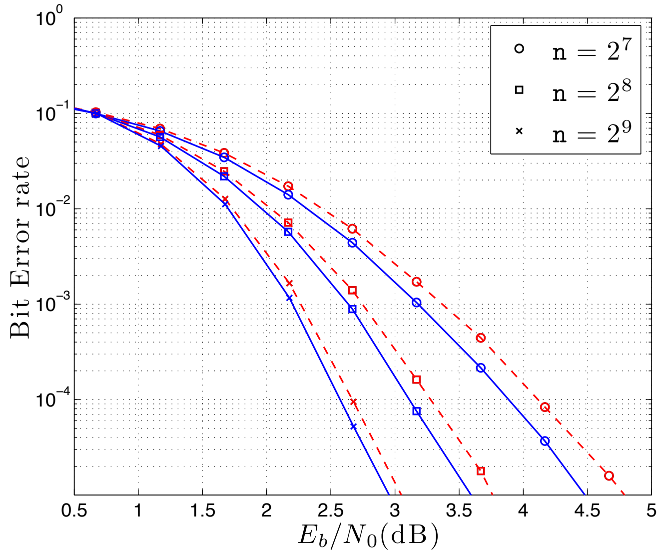
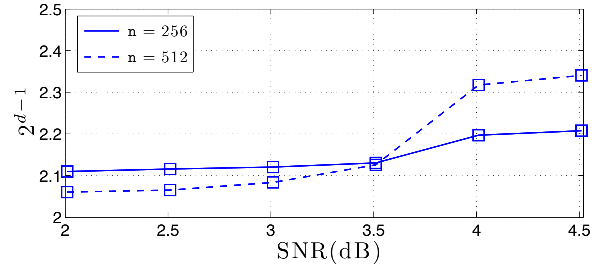


Fig. 4: BER over the AWGN channel for the BP (dashed lines) and the TEP (solid lines) decoders for the irregular LDPC code and different code-lengths n, 128(○), 256(□) and 512(×).

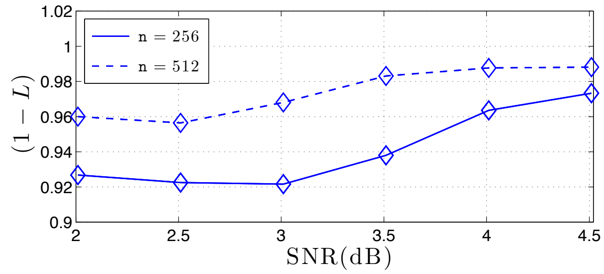
different kind of trees/forests for each receive word, providing measurable gains in the finite-length regime. Taking into account that the complexity measures show that the TEP decoder only increases slightly the BP cost, the TEP decoder postulates as a competitive technique for practical finite-length scenarios.

VII. Acknowledgments

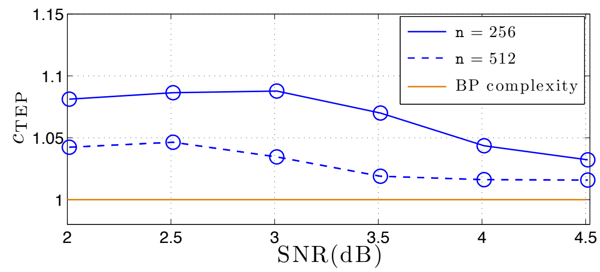
This work was partially funded by Spanish government (Ministerio de Educación y Ciencia, TEC2012-38800-C03–01/02, Consolider-Ingenio 2010 CSD2008-00010), Universidad Carlos III de Madrid/Comunidad de Madrid (CCG10-UC3M/TIC-5304) and European Union (FEDER).



(a)



(b)



(c)

Fig. 5: Complexity measures for the TEP decoder, for $n = 256$ (solid lines) and $n = 512$ (dashed lines).

References

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

[2] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible Inference*. Morgan Kaufmann, 1988.

[3] D. MacKay and R. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electronics Letters*, vol. 32, no. 18, p. 1645, Aug. 1996.

[4] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.

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

[6] M. Lentmaier, A. Sridharan, D. Costello, and K. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. on Inf. Theory*, vol. 56, no. 10, pp. 5274 –5289, Oct. 2010.

[7] S. Kudekar, T. Richardson, and R. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," in *IEEE Int. Symp. on Inf. Theory Proc. (ISIT)*, June 2010, pp. 684 –688.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Trans. on Inf. Theory*, vol. 51, pp. 2282–2312, 2004.

[9] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*. Foundations and Trends in Machine Learning, 2008.

[10] T. Minka and Y. Qi, "Tree-structured approximations by expectation propagation," in *Proc. of the Neural Inf. Process. Syst. Conf., (NIPS)*, 2003.

[11] P. Pakzad and V. Anantharam, "Kikuchi approximation method for joint decoding of LDPC codes and partial-response channels," *IEEE Trans. on Commun*, vol. 54, no. 7, pp. 1149 –1153, July 2006.

[12] T. Minka, "Expectation propagation for approximate Bayesian inference," in *UAI*, 2001, pp. 362–369.

[13] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for LDPC tree-structure expectation propagation for LDPC decoding over the BEC," *Accepted for publication in IEEE Trans. on Inf. Theory*, 2013. [Online]. Available: http://arxiv.org/abs/1009.4287

[14] L. Salamanca, P. M. Olmos, J. J. Murillo-Fuentes, and F. Perez-Cruz, "Tree expectation propagation for ML decoding of LDPC codes over the BEC," *IEEE Trans. on Commun*, p. Forthcoming, 2013.

[15] D. Burshtein and G. Miller, "Efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. on Inf. Theory*, vol. 50, no. 11, pp. 2837 – 2844, Nov. 2004.

[16] A. Shokrollahi and M. Luby, *Raptor Codes*, ser. Found. and trends in Comm. and Inf. theory. Now Publishers Inc, 2011, vol. 6, no. 3-4.

[17] A. Becker, R. Bar-Yehuda, and D. Geiger, "Randomized algorithms for the loop cutset problem," *Journal of Artificial Intelligence Research*, vol. 12, no. 1, pp. 219–234, 2000.

[18] O. Shental, A. J. Weiss, N. Shental, and Y. Weiss, "Generalized belief propagation receiver for near-optimal detection of two-dimensional channels with memory," in *IEEE Inf. Theory Workshop (ITW)*, 2004.