Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 92659, 9 pages
doi:10.1155/2007/92659

Research Article

Enhancement of Unequal Error Protection Properties of LDPC Codes

Charly Poulliat, David Declercq, and Inbar Fijalkow

ETIS laboratory, UMR 8051-ENSEA/UCP/CNRS, 6 Avenue du Ponceau, 95014 Cergy-Pontoise, France

Received 13 March 2007; Revised 19 August 2007; Accepted 2 October 2007

Recommended by Michael Gastpar

It has been widely recognized in the literature that irregular low-density parity-check (LDPC) codes exhibit naturally an unequal error protection (UEP) behavior. In this paper, we propose a general method to emphasize and control the UEP properties of LDPC codes. The method is based on a hierarchical optimization of the bit node irregularity profile for each sensitivity class within the codeword by maximizing the average bit node degree while guaranteeing a minimum degree as high as possible. We show that this optimization strategy is efficient, since the codes that we optimize show better UEP capabilities than the codes optimized for the additive white Gaussian noise channel.

1. INTRODUCTION

The subject of the paper is the enhancement of unequal error protection (UEP) properties of low-density parity-check (LDPC) codes suitable for UEP transmission schemes (scalable image transmission, etc.). UEP transmission schemes are considered to take into account the different error sensitivities of the source bitstream. For example, semantic information like headers have to be almost error free to avoid the crash of the source decoding, whereas the data symbols can usually tolerate some errors. The UEP can significantly enhance the performance of the global communication system compared to an equal error protection scheme since the overall redundancy is allocated in order to provide more protection to the most sensitive parts of the bitstream.

Apart from multilevel coded modulations, there are commonly two ways to provide unequal protection depending on the transmission scheme we consider. Firstly, one can adapt the protection level by adaptively changing the code rate through puncturing. Some methods for an efficient design of such adaptive coding schemes are proposed in the literature either for punctured convolutional codes [1], for punctured or pruned turbocodes [2, 3], or for punctured LDPC codes [4, 5]. As these schemes are implemented using a unique mother code, which is punctured depending on the required level of protection, they have been widely proposed in combined source and channel coding (CSCC) schemes for the purpose of multimedia transmissions (see, e.g., [6] and references therein). In these UEP coding schemes, rate-compatible error correcting schemes are combined with a rate-allocation algorithm to mainly minimize the distortion of the source at the receiver. As the channel code design and UEP allocation algorithm are considered *separately*, making the design of good rate compatible coding schemes completely independent of the considered CSCC scheme, these coding schemes can be further improved by considering a joint source-channel (de)coding approach (see [7, 8], e.g., in case of LDPC codes). In these approaches, the distortion-rate or mutual information transfer function are directly used for the optimization of the irregularity profiles, and thus, UEP allocation could naturally arise from the optimization process.

Secondly, one can design codes that provide inherent unequal error protection within a codeword. This was early studied in [9, 10] to design linear UEP codes. For such codes, the bits within a codeword do not have the same protection level and error correction capability. For example, if we consider packet-based transmissions with no possibility of retransmission, an error in network/transport layer headers is much more critical than an error in the payload to avoid the rejection of the packet. Moreover, for some wireless video-based communications, syntax reordering mechanisms, such as data partitioning, are used to reorder data in the bitstream in accordance with their semantic importance (e.g., I/P/B frames or movement vectors) and their required level of protection. In all these cases, for a given source codeword, the most sensitive bits are associated to information bit positions with the highest protection level.

In this context, irregular LDPC codes can provide *inherent* unequal error capability due to the different connection degrees of the bit nodes. As widely observed, it appears that highly connected nodes are more protected than weakly connected ones. This irregularity can be used to provide inherent UEP, in the sense that high sensitivity bits are associated with high connection degree bit nodes [11, 12]. The optimization of the LDPC irregularity has already been proposed for different channels as the binary erasure channel (BEC) [13] or the additive white Gaussian noise (AWGN) channel [14]. In these approaches, the codeword irregularity is globally optimized for a specific channel considering therefore that each and every bit in the codeword has the same average error probability. Thus, the existing optimization techniques do not necessarily ensure a good UEP capability, as underlined in [15]. The work in [15] first proposed an ad-hoc graph-based approach to optimize the irregularity profile in order to have UEP within a codeword with two classes of sensitivity. This work has been more formally stated and extended by [16–18], for the design of UEP LDPC codes for the BEC: considering two information classes, they first derive the density evolution for the BEC channel using a more refined parametrization to take into account the UEP characterization. Then, they optimize the irregularity profile in order to improve the performance of the most sensitive class; but they only consider partially regular (same variable node degree per class), leading to a limited solution search space. More recently, the studies for the UEP characterization of LDPC codes have been extended to the case of Raptor codes [19]; but however, no other channel than the BEC has been considered so far for the characterization of the UEP behavior of LDPC codes. All these approaches are mainly based on asymptotic assumptions to derive good irregularity profiles, even if they present some algebraic construction aspects for finite-length design [17]. Apart from these contributions, LDPC-based algebraic constructions for UEP applications were proposed. A simple scheme based on lower block triangular parity-check matrix has been proposed in [20] with a two-stage decoding algorithm. The UEP property arises directly from the simple structure of the designed code. A more refined algebraic construction has been proposed in [21], where a layered Plotkin-based construction is proposed based on LDPC component codes. In this scheme, the optimization consists of selecting good component codes, and UEP arises from the structure of the layered coding scheme. Note that for that kind of construction, multistage decoding (or an improved version) is preferred to belief propagation decoding due to the inherent presence of short cycles in the graph of the code.

As opposed to these methods, using asymptotic tools, we investigate the characterization of UEP for LDPC codes for the Gaussian channel by allowing parts of the same codeword to have their own irregularity profiles, regardless of particular algebraic structures. In this paper, we show that it is possible to generalize the asymptotic characterization and optimization techniques in order to take into account a target UEP behavior of the LDPC code for the Gaussian channel. As discussed in details in Section 4, we interpret the UEP properties of an LDPC code as different local convergence speeds. Actually, the most protected class will be assigned to the bits in the codeword which converge to their right value in the minimum number of decoding iterations.

The new design strategy that we propose is based on two main differences compared to existing work. Firstly, we show that by appropriately changing the optimization objective function and by adding constraints, it is possible to increase the local convergence speed, and then to enhance the UEP properties of the LDPC code without significantly degrading the overall error convergence performance. Secondly, we use an *extended parametrization* for the LDPC code irregularity, that is adapted to a desired UEP scheme, by allowing distinct irregularity profiles for each sensitivity class. We then propose a hierarchical optimization procedure to design efficiently an LDPC code irregularity in this framework.

The paper is organized as follows. In Section 2, we describe the UEP parameters and give some notations used in the paper. In Section 3, we analyze how the irregularity can be exploited for UEP and derive a cost function suitable for UEP. The optimization algorithm we propose is given in Section 4. Optimization results and finite-length simulations are given and interpreted in Section 5. Conclusions and perspectives are drawn in Section 6.

2. UEP PARAMETER DESCRIPTION AND ASSOCIATED LDPC PARAMETRIZATION

2.1. UEP parameter description and notations

The transmission scheme consists of sending a coded bitstream under given UEP constraints over the AWGN channel with binary input and noise variance parameter σ^2. The UEP-coded bitstream can be described and parameterized as follows: let a channel codeword of a rate R LDPC code be divided into N_c classes ordered in decreasing order of their error sensitivity. Thus, considering the set of N_c classes $\{C_k \mid k = 1 \dots N_c\}$, C_1 will be associated with the highest required protection level and C_{N_c} with the lowest. The redundancy bits of the channel codeword are associated with class C_{N_c} and the information bits are associated with the $(N_c - 1)$ first classes. Let the proportions $\underline{\alpha} = \{\alpha_k \mid k = 1 : N_c - 1\}$ be the normalized lengths of each class corresponding to the information bits with $\sum_{k=1}^{N_c-1} \alpha_k = 1$. The proportions $\underline{\alpha}$ are usually provided by a data partitioning mechanism, for example. The proportions of bits in the channel codeword belonging to the different classes $\{C_k / k = 1 \dots N_c\}$ are given by $\underline{p} = \{\alpha_1 R, \dots, \alpha_{N_c-1} R, (1 - R)\}$.

Example

We consider a rate $R = 1/2$ code and three sensitivity classes. The redundancy bits are associated with class C_3 and the information bits are divided into two error sensitivity classes: C_1 is the most sensitive class (typically headers) and C_2 is the less sensitive class (typically data). The proportion distribution for information bits is given by $\underline{\alpha} = \{\alpha, 1 - \alpha\}$. The proportions of bits of the channel codeword belonging to the different classes $\{C_1, C_2, C_3\}$ are given by $\underline{p} = \{\alpha R, (1-\alpha)R, (1-R)\}$.

Let $d_{c_{\max}}$ and $\rho(x) = \sum_{j=2}^{d_{c\max}} \rho_j x^{j-1}$ be the maximum check node connection degree and the generating polynomial of the proportion of edges connected to check nodes with connection degree j [22], respectively. We assume that $\rho(x)$ is the same for each class. Let $d_{v_{\max}}^{(k)}$ be the maximum bit node connection degree in the class C_k. For each class C_k, we define $\lambda^{(C_k)}(x) = \sum_{i=2}^{d_{v\max}^{(k)}} \lambda_i^{(C_k)} x^{i-1}$ and $\widetilde{\lambda}^{(C_k)}(x) = \sum_{i=2}^{d_{v\max}^{(k)}} \widetilde{\lambda}_i^{(C_k)} x^{i-1}$ the generating polynomial of the proportion of edges connected to bit nodes with connection degree i and the dual generating polynomial, where $\widetilde{\lambda}_i^{(C_k)}$ is the fraction of degree-i bit nodes. The following equalities hold:

$$\begin{gathered} \sum_{k=1}^{N_c} \lambda^{(C_k)}(1) = 1, \qquad \sum_{k=1}^{N_c} \widetilde{\lambda}^{(C_k)}(1) = 1; \\ \forall k = 1, \ldots, N_c - 1, \quad \sum_{i=2}^{d_{v\max}^{(k)}} \widetilde{\lambda}_i^{(C_k)} = \alpha_k R; \\ \sum_{i=2}^{d_{v\max}^{(k)}} \widetilde{\lambda}_i^{(N_c)} = (1-R). \end{gathered} \tag{1}$$

The relationbetween $\lambda_i^{(C_k)}$ and $\widetilde{\lambda}_i^{(C_k)}$ is given by

$$\widetilde{\lambda}_i^{(C_k)} = \frac{\lambda_i^{(C_k)}/i}{\sum_k \sum_{i'} \lambda_{i'}^{(C_k)}/i'}. \tag{2}$$

In the sequel, we denote $\underline{\lambda}^{(C_k)} = [\lambda_2^{(C_k)}, \ldots, \lambda_{d_{v\max}^{(k)}}^{(C_k)}]^\top$ and $\underline{\rho}$ the vectors associated with $\lambda^{(C_k)}(x)$ and $\rho(x)$, respectively. $\mathbf{1}$ is a one valued vector and $\top$ is used for the transpose vector. We assume that $d_{v_{\max}} = \max(d_{v_{\max}}^{(k)}) \forall k = 1, \ldots, N_c$. We also set $\underline{1/d_v} = [1/2, \ldots, 1/d_{v_{\max}}]^\top$, $\underline{1/d_c} = [1/2, \ldots, 1/d_{c_{\max}}]^\top$, and $\underline{\lambda} = [\underline{\lambda}^{(C_1)}, \ldots, \underline{\lambda}^{(C_{N_c})}]^\top$. With these notations, an LDPC code irregularity is parameterized by $(\underline{\lambda}, \underline{\rho}, \underline{p})$.

Note that this detailed parametrization of the bit node irregularity is specifically matched to several classes of different sensitivities and is a necessary step for LDPC code optimization under UEP constraints. By allowing parts of the same codeword to have their own irregularity profile, it is possible to exhibit local behavior for the belief propagation (BP) decoder and the adapted optimization strategy described in Section 4 capitalizes on this detailed parametrization in order to yield LDPC codes with enhanced UEP properties.

2.2. Mutual information evolution

In order to optimize LDPC codes, we need analytical tools to study the convergence of the LDPC code depending on of the code parameters. Using a Gaussian assumption for log density ratio (LDR) message and independence assumption between LDR messages (infinite codeword size), we can explicitly give the evolution of the mutual information (MI) associated with the mean of the LDR messages for one decoding iteration depending on the code parameters when BP decoding is used [23]. The explicit relation describing the evolution of the MI from iteration $l-1$ to iteration l defines the EXIT chart associated with the LDPC code.

We denote $x_u^{(l)}$ and $x_v^{(l)}$, the mutual information associated with LDR messages at the input of bit nodes and the mutual information associated with LDR messages at the input of check nodes at the lth decoding iteration, respectively.

Assuming Gaussian approximation [14, 23], we have:

(1) check node message update:

$$x_u^{(l-1)} = 1 - \sum_{j=2}^{d_{c\max}} \rho_j \mathbf{J}\big((j-1)\mathbf{J}^{-1}(1 - x_v^{(l-1)})\big), \tag{3}$$

(2) bit node message update:

$$x_v^{(l)} = \sum_{k=1}^{N_c} \sum_{i=2}^{d_{v\max}} \lambda_i^{(C_k)} \mathbf{J}\left(\frac{2}{\sigma^2} + (i-1)\mathbf{J}^{-1}(x_u^{(l-1)})\right), \tag{4}$$

with $J(\cdot)$ being the mutual information function $J(m) = 1 - \mathbb{E}(\log_2(1+e^{-x}))$ of a Gaussian random variable $x \sim \mathcal{N}(m, 2m)$. Combining (3) and (4) gives the EXIT chart of the LDPC code:

$$x_v^{(l)} = F(\underline{\lambda}, x_v^{(l-1)}, \sigma^2). \tag{5}$$

For more details on LDPC EXIT charts refer to [23]. The initial condition is given by $x_v^{(0)} = 0$. The condition $F(\underline{\lambda}, x, \sigma^2) > x \forall x \in [0,1)$ ensures the convergence of BP algorithm to an error-free codeword.

3. COST FUNCTION FOR LDPC OPTIMIZATION WITH UEP CONSTRAINTS

In this section, we discuss and analyze how the irregularity of the LDPC code can be used and optimized to provide UEP.

3.1. Providing UEP with capacity achieving codes: a valid question ?

LDPC codes exhibit a threshold behavior depending on the channel signal-to-noise ratio E_b/N_0: above a given E_b/N_0 threshold δ, the word-error probability P_w is zero as the word length N tends to infinity. Let us remark at this point that the asymptotic performance criterion for optimizing families of LDPC code is the gap between the convergence threshold and the Shannon limit defined as *a zero-frame error rate* [24]. As a consequence, the optimized LDPC codes with large code lengths cannot provide any unequal error protection since the goal is to have no errors at all in the codeword. This complicates the task of providing UEP behavior with LDPC codes. On the other hand, more and more standards use modern coding schemes like LDPC or Turbocodes, which means that looking for UEP capabilities with these codes is indeed a key research issue. A general discussion about this issue is done in this section.

When using the asymptotic characterization of the preceding section, one assumes that all edges are sequentially updated at the bit node side and the check node side. This particularly means that the EXIT chart equations express only the global behavior of one decoding iteration and no local behavior is taken into account. With this approach, there is no possibility to provide controlled unequal error protection, which is by nature a local property.

Irregular LDPC codes are however codes which are naturally well suited for UEP if we consider the different protection levels as parts of the codeword which converge more rapidly than others. In an irregular LDPC code, some bits are more protected than others *after a single decoding iteration* since the connectivity differs from one bit to another. For example, if a bit is connected to a large number of edges, sometimes denoted as *elite bit*, it gets a lot of information in a single iteration while a low connected bit receives less information and will be less protected.

So even with asymptotic analysis, it is possible to have UEP LDPC codes with the effect of different *local convergence speeds*. The general idea is then to have the most sensitive class which converges the most rapidly in order to have it error free with the minimum number of iterations. Of course, the difference in error protection will be diminished with an increased number of decoding iterations, and eventually vanish with an infinite number of iterations.

Let us then illustrate the link between bit node irregularity and the associated bit error probability after a finite number of iterations. If we assume that the graph is locally a tree (in the sense of [24]), the bit error probability at the lth decoding iteration for bit nodes with a connection degree i under Gaussian approximation [14] is given by

$$P_i^{(l)} = Q\left(\sqrt{\frac{(2/\sigma^2) + i\mathbf{J}^{-1}(x_u^{(l)})}{2}}\right), \tag{6}$$

where $Q(\cdot)$ is the Gaussian tail function given by

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{+\infty} e^{-z^2/2}dz. \tag{7}$$

Above a threshold δ, $\mathbf{J}^{-1}(x_u^{(l)})$ is strictly increasing with decoding iteration l. $Q(\cdot)$ being a strictly decreasing function, (6) shows that, at a given iteration l, the more connected a bit node is, the more "protected" it is, in the sense that it has a smaller error probability. This also implies that the convergence is faster for the highly connected bits. Now that we have explained how the irregularity can help us to provide UEP, we focus on the case of a specific class of sensitivity in a coded bitstream.

3.2. Protection within a single class of sensitivity

As defined in Section 2.1, the codeword is divided into N_c sensitivity classes. For a given code profile, each and every class C_k within the codeword is associated with a dual parametrization polynomial $\tilde{\lambda}^{(C_k)}(x) = \sum_{i=2}^{d_{v_{\max}}} \tilde{\lambda}_i^{(C_k)} x^{i-1}$.

According to [14], the bit error probability $P_l^{(C_k)}$ for the class C_k is defined as

$$P_l^{(C_k)} = \frac{1}{\alpha_k R}\sum_{i=d_{v_{\min}}^{(k)}}^{d_{v_{\max}}} \tilde{\lambda}_i^{(C_k)} Q\left(\sqrt{\frac{(2/\sigma^2) + i\mathbf{J}^{-1}(x_u^{(l)})}{2}}\right). \tag{8}$$

$d_{v_{\min}}^{(k)}$ is the minimum bit node degree in the class C_k. From (8), we can derive a lower and an upper bound on the bit error probability in class C_k (see (9)). The lower bound is obtained using convexity arguments on the $Q(\cdot)$ function and the upper bound is obtained using the decreasing property of the $Q(\cdot)$ function. Note that these bounds are in general very loose and we will use them only to justify our approach:

$$\begin{aligned} &Q\left(\sqrt{\frac{(2/\sigma^2) + \overline{\tilde{\lambda}^{(C_k)}}\mathbf{J}^{-1}(x_u^{(l)})}{2}}\right) \\ &\quad \le P_l^{(C_k)} \le Q\left(\sqrt{\frac{(2/\sigma^2) + d_{v_{\min}}^{(k)}\mathbf{J}^{-1}(x_u^{(l)})}{2}}\right). \end{aligned} \tag{9}$$

In (9), the average bit node degree associated with the class C_k is defined as

$$\overline{\tilde{\lambda}^{(C_k)}} = \frac{1}{\alpha_k R}\sum_{i=d_{v_{\min}}^{(k)}}^{d_{v_{\max}}} \tilde{\lambda}_i^{(C_k)} i. \tag{10}$$

The lower bound corresponds to the case where we consider the bit error probability associated with the mean of the *a posteriori* LDR messages associated with class C_k using a Gaussian assumption. The upper bound is the limit case of a uniform connection degree distribution associated with class C_k, as if the LDPC code would be regular with degree $d_{v_{\min}}^{(k)}$ in the class.

According to (9), the bit error probability is closely related to $\overline{\tilde{\lambda}^{(C_k)}}$ and $d_{v_{\min}}^{(k)}$. However, it is not easy to track the behavior of the bit error probability with (9) since the MI at the lth iteration $x_u^{(l)}$ is a function of both quantities $\overline{\tilde{\lambda}^{(C_k)}}$ and $d_{v_{\min}}^{(k)}$. To circumvent this dependance, we make the assumption that two codes with close thresholds have almost the same convergence rate (the evolution of the quantities $x_u^{(l)}$ are very close), regardless of the parameters values $\overline{\tilde{\lambda}^{(C_k)}}$ and $d_{v_{\min}}^{(k)}$.

Based on this assumption, we can study the influence of $\overline{\tilde{\lambda}^{(C_k)}}$ and $d_{v_{\min}}^{(k)}$ on the bit-error probability:

(1) for a given $\overline{\tilde{\lambda}^{(C_k)}}$, by maximizin $d_{v_{\min}}^{(k)}$, we force the inequality to be as tight as possible;

(2) by setting $\overline{\tilde{\lambda}^{(C_k)}}$ as large as possible, we try to minimize the minimum bit-error probability we can reach for this class.

This implies that, by jointly optimizing the average bit node degree and the minimum bit node degree associated with a class, we expect to obtain a bit-error probability as small as possible for a fixed and relatively small number of decoding iterations. Therefore, for a given class C_k, we propose to use as a cost function the maximization of the average bit node degree $\overline{\tilde{\lambda}^{(C_k)}}$ subject to a maximum bit node degree $d_{v_{\min}}^{(k)}$.

Inserting (2) in (10) and considering that the code rate R is constant (and so, $\sum_k \sum_{i'} \lambda_{i'}^{(C_k)}/i'$ is constant), the maximization of (10) is equivalent to the maximization of

$$\sum_{i=d_{v_{\min}}^{(k)}}^{d_{v_{\max}}} \lambda_i^{(C_k)}. \tag{11}$$

This can be interpreted as a maximization of the proportion of edges in the part of the parity check matrix associated with the the class C_k. Thus, more messages will transit to this part of the graph ensuring a faster local convergence.

4. A HIERARCHICAL APPROACH FOR LDPC CODES OPTIMIZATION WITH UEP CONSTRAINTS

In this section, we present our optimization strategy to enhance UEP properties of LPDC codes when transmission over the AWGN channel is considered. Based on the previous observations, we show that the optimization problem can be solved through a hierarchical process, each step consisting of the optimization of the average bit node degree in a single class subject to a maximized minimum degree and some constraints provided by previous steps. Each step can be achieved by linear programming.

According to [14, 22], we will consider the LDPC codes that converge to a vanishing bit error probability at a given E_b/N_0 (the *code threshold*). Let us denote δ the threshold of the optimized LDPC irregularity *without* UEP constraints as in [14]. An optimization algorithm that takes into account the specific UEP constraints will result in an optimized code with a threshold greater than δ (worse threshold).

In order to be sure that the UEP constraints do not lead to a too-large degradation of the threshold, we limit the set of possible LDPC codes to those whose convergence threshold lies within $[\delta, \delta + \epsilon]$, with ϵ a small constant fixed in the optimization algorithm. If ϵ is small enough, the global convergence of the code will be approximatively the same as for the code obtained without UEP constraints, as explained in Section 3.2.

4.1. *The optimization of a class profile as a conditional linear programming problem*

In this section, we only focus on the optimization of a single class C_k, that is, the optimization of the irregularity for the part of the codeword associated with this class. We assume that all the optimizations for classes $\{C_{k'}, k' < k\}$ have already been performed and the results of these optimizations are used as constraints in the current optimization process. At the beginning of the optimization of a single class C_k, we assume that the following parameters are given: $d_{v_{\max}}$, $\rho(x)$, δ, ϵ, $\underline{\alpha}$, and $\underline{\lambda}^{(C_{k'})}$ $\forall k' < k$ (The last parameters are assumed to be known from previous optimization steps).

The proposed optimization is performed by maximizing the average bit node degree of the class C_k for a decreasing $d_{v_{\min}}^{(k)}$ from $d_{v_{\max}}$ to 2. The iterative procedure is stopped when a solution of an LDPC code is found which converges at a fixed threshold $\in [\delta, \delta + \epsilon]$. For a given threshold $\delta + \epsilon$ (and then a given noise power σ^2) and a check node degree distribution $\rho(x)$, the optimization of the class C_k can be stated as a linear programming problem subject to three types of constraints as shown in Algorithm 1.

(1) Initialization $d_{v_{\min}}^{(k)} = d_{v_{\max}}$.
(2) While optimization failure (any constraint is not fulfilled):
  (a) maximize the average bit node degree (cf. (10) and (11))

$$\max_{\underline{\lambda}} \underline{\lambda}^{(C_k)^\top} 1 \tag{12}$$

under the following constraints
- global constraints:

$[C_1]$ rate constraint:

$$\sum_k \underline{\lambda^{(C_k)}}^\top \underline{1/d_v} = (1-R)^{-1} \underline{\rho}^\top \underline{1/d_c}, \tag{13}$$

$[C_2]$ proportion constraints:

(i)
$$\sum_k \underline{\lambda}^{(C_k)^\top} 1 = 1, \tag{14}$$

(ii)
$$\forall k \in \{1, \ldots, N_c - 1\},\quad \underline{\lambda}^{(C_k)^\top} \underline{1/d_v} = \alpha_c \frac{R}{1-R} \underline{\rho}^\top \underline{1/d_c}, \tag{15}$$

$[C_3]$ convergence constraint (cf. (5)):

$$\mathrm{F}(\underline{\lambda}, x, \sigma^2) > x, \tag{16}$$

$[C_4]$ stability condition:

$$\sum_k \lambda_2^{(C_k)} < e^{1/2\sigma^2} / \sum_{j=2}^{d_{c_{\max}}} \rho_j (j-1), \tag{17}$$

- class constraint:

$[C_5]$ minimum bit node degree constraint:

$$\forall i < d_{v_{\min}}^{(k)}, \lambda_i^{(C_k)} = 0, \tag{18}$$

- conditional constraints:
- $[C_6]$ Previous optimizations constraints:

$$\forall k' < k, \underline{\lambda}^{(C_{k'})} \text{ is fixed} \tag{19}$$

  (b) $d_{v_{\min}}^{(k)} = d_{v_{\min}}^{(k)} - 1$
end

Algorithm 1

When the optimization process is successful, we store the distribution associated to the class being optimized in order to use it as a constraint for the next class.

The cost function used in (12) only depends on $\underline{\lambda}^{(C_k)}$, which is a cost function only afferent to the class to be optimized. The optimization results are however the vectors $\{\underline{\lambda}_{\text{opt}}^{(C_{k'})} \forall k' \geq k\}$ which are involved in the global constraints.

(1) Choose $E_b/N_0 = \delta + \epsilon$.
(2) For $k = 1 \dots N_c - 1$:
  a) find $\underline{\lambda}_{\text{opt}}^{(C_k)}$ and $d_{v_{\min}\,\text{opt}}^{(k)}$ with the optimization procedure described in Section [4.1].
  b) compute the constraints of the next step with $\{(\underline{\lambda}_{\text{opt}}^{(C_{k'})}, d_{v_{\min}\,\text{opt}}^{(k')})\,\forall k' \leq k\}$

Algorithm 2: Hierarchical UEP optimization algorithm.

The conditions $[C_1]$–$[C_4]$ are global constraints related to code convergence, rate, and proportion distribution constraints. $[C_5]$ is related to the local constraint of the minimum bit node degree of the class C_k; and finally $[C_6]$ takes into account the optimized irregularities of the previous classes.

4.2. *Hierarchical optimization algorithm*

We propose a hierarchical approach for the successive optimization of all classes. We will start to optimize the most sensitiveclass and perform the hierarchical optimization in decreasing order of sensitivity. Assuming that $d_{v_{\max}}$ and $\rho(x)$ are given, Algorithm 2 illustrates this hierarchical approach.

Optimizing the classes in decreasing order of sensitivity tries to take into account that the source data are partitioned in decreasing order of sensitivity and that the source decoding is usually sequential. In the following section, we will apply this algorithm with different values of ϵ and compare to an LDPC code optimized for the AWGN channel without UEP constraints.

5. RESULTS

In this section, some simulation results are presented to illustrate the performance of the LDPC codes optimized for the AWGN channel with UEP constraints. First, we analyze the results provided by the linear programming optimization in the case of infinite codeword length. Then, we focus on the performance in the case of finite codeword length.

5.1. *Influence of threshold offset on the code irregularity*

We consider a UEP transmission scheme with 3 classes within a codeword: C_1 is the high error sensitivity information bits class, C_2 the low error sensitivity information bits class, and C_3 is assigned to redundancy bits. The information bits proportions are given by $\underline{\alpha} = (\alpha_1, \alpha_2)$, which can take different values. We will consider rate $R = 1/2$ LDPC codes. We assume that $d_{v_{\max}}$ is fixed to $d_{v_{\max}} = 30$. $\rho(x) = 0.0437x^7 + 0.9563x^8$ is fixed to the value of the AWGN optimized LDPC code [25]. δ is the E_b/N_0 threshold in dB of the AWGN optimized code and ϵ is the E_b/N_0 offset.

Figure [1] gives the minimum degree for the first class versus ϵ for different information bit proportions $\underline{\alpha} = (\alpha_1, \alpha_2)$. As we can see, the minimum degree increases with

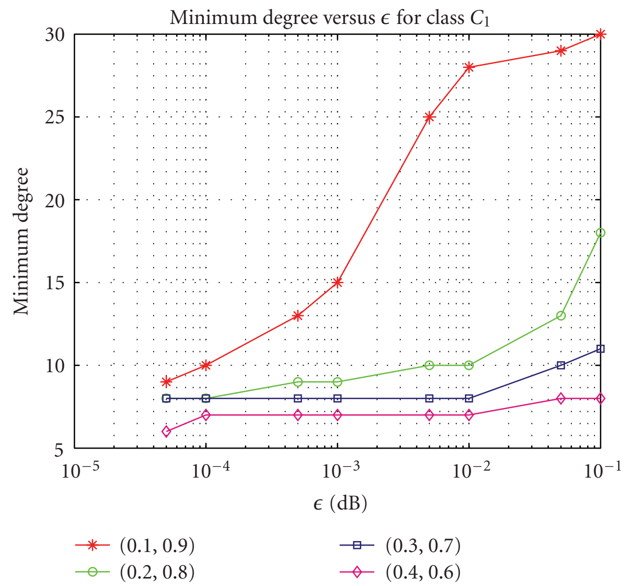

Figure 1: Minimum bit node degree for class 1 versus ϵ. As expected, the minimum bit node degree increases with ϵ. The increase is faster when proportions associated with the class 1 are low.

Table 1: Degree distributions for $R = 1/2$ AWGN code with $\underline{\alpha} = (0.3, 0.7)$. The higher connection degrees are associated with the more sensitive class as done in [11].

AWGN					
C_1		C_2		C_3	
λ_7	0.0271	λ_3	0.1765	λ_2	0.2114
λ_8	0.1587	λ_5	0.0541	λ_3	0.0180
λ_{30}	0.2943	λ_7	0.0599	—	—
$\sum_i \lambda_i^{(C_1)}$	0.4801	$\sum_i \lambda_i^{(C_2)}$	0.2905	$\sum_i \lambda_i^{(C_3)}$	0.2294

increasing ϵ values. Therefore, the UEP-LDPC code is expected to exhibit a better UEP behavior than the AWGN optimized code for this class; as for low ϵ values, the minimum degree converges to the minimum degree of the AWGN optimized code. We also remark that the increase of the minimum degree is less significant when $\underline{\alpha}$ corresponds to a uniform repartition.

Tables 1, 2 and 3 give some degree distributions for $\underline{\alpha} = (0.3, 0.7)$, $d_{v_{\max}} = 30$ and $\rho(x) = 0.0437x^7 + 0.9563x^8$. As we can see, when ϵ increases, the minimum degree of the first class is increasing and the total edge proportion associated to the first class is maximized. For the second class, we can see the effect of the hierarchical procedure: we have less diversity in terms of connection degrees associated with the second class. This results in a concentration of the connection degrees of the second class on lower connection degrees.

In order to predict the behavior of the expected gain achieved by the UEP-LDPC codes, we study the asymptotic E_b/N_0 gains achieved for class 1 and 2 with $l = 7$ decoding iterations and bit error rate BER $= 10^{-5}$ when using MI evolution. In order to compare UEP capabilities of the UEP-LDPC

Table 2: Degree distributions for the different classes with $\epsilon = 0.05$. $R = 1/2$.

UEP ($\epsilon = 0.05$)					
C_1		C_2		C_3	
λ_{10}	0.2310	λ_3	0.14615	λ_2	0.2100
λ_{11}	0.0218	λ_4	0.11795	λ_3	0.0201
λ_{30}	0.2529	—	—	—	—
$\sum_i \lambda_i^{(C_1)}$	0.5058	$\sum_i \lambda_i^{(C_2)}$	0.2641	$\sum_i \lambda_i^{(C_3)}$	0.2301

Table 3: Degree distributions for the different classes with $\epsilon = 0.5$. $R = 1/2$.

UEP ($\epsilon = 0.5$)					
C_1		C_2		C_3	
λ_{16}	0.4774	λ_3	0.2346	λ_2	0.2210
λ_{17}	0.0573	—	—	λ_3	0.0036
λ_{18}	0.0027	—	—	—	—
λ_{19}	0.0010	—	—	—	—
λ_{20}	0.0024	—	—	—	—
$\sum_i \lambda_i^{(C_1)}$	0.5408	$\sum_i \lambda_i^{(C_2)}$	0.2346	$\sum_i \lambda_i^{(C_3)}$	0.2246

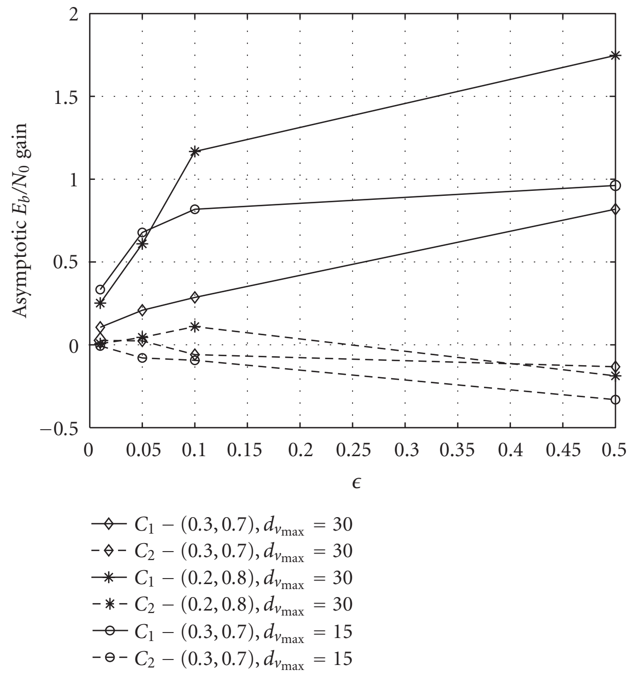

Figure 2: E_b/N_0 gain for information classe C_1 and C_2. The E_b/N_0 gain is obtained for BER $= 10^{-5}$.

codes and the AWGN optimized LDPC codes, we assign for the AWGN optimized code the information bits belonging to the first class to the $\alpha_1 R$ most connected bit nodes, the information bits of the class C_2 to the $\alpha_2 R$ most connected remaining bit nodes, and so on up to the $N_c^{\text{th}} - 1$ class. Finally, the redundancy bits are associated to the remaining $(1 - R)$ bit nodes. This is the natural way of assigning bit nodes to provide enhanced UEP capabilities to the first classes.

The study is done for various $\underline{\alpha}$ and $d_{v_{\max}} = 30$ and $d_{v_{\max}} = 15$ and $R = 1/2$. For $d_{v_{\max}} = 15$, we have the following parameters $\rho(x) = x^7$ and $\underline{\alpha} = (0.3, 0.7)$. The simulation results are given in Figure 2.

The asymptotic E_b/N_0 gain is defined as the difference in E_b/N_0 to reach a BER$= 10^{-5}$ (8) between the LDPC codes optimized with UEP constraints and the LDPC code, optimized for the AWGN channel without UEP constraints. In all cases, the E_b/N_0 gain for the first class increases when ϵ increases. In that sense, we have improved the UEP capability for the first class. For the second class, the UEP behavior depends on ϵ. For small values of ϵ and $d_{v_{\max}} = 30$, the second class exhibits also a slight improvement compared to the AWGN optimized code. However, when ϵ increases, the UEP-LDPC codes perform worse than the AWGN code for the second class. This can be interpreted as follows: when ϵ increases, since we first optimize the more sensitive class, we assign too many edges to the first class. The second class will be more constrained and will have concentrated low degrees. Moreover, with increasing ϵ, the global convergence of the UEP-LDPC codes become worse, and since the second class is weaker, it is more sensitive to convergence weakness. For a given $\underline{\alpha}$, we observe that, for $d_{v_{\max}} = 15$, we reach the best UEP capability faster than for $d_{v_{\max}} = 30$.; but this has to be balanced with the performance of the second class: the asymptotic E_b/N_0 loss is less for $d_{v_{\max}} = 30$ than for $d_{v_{\max}} = 15$. Finally, it can be seen that for a given $d_{v_{\max}}$, the gain can be better for $\underline{\alpha}$ that are far from uniform distribution, since we can allocate more edges for the first class.

All these observations are confirmed by simulations. Hence, in order to select a good candidate for improved UEP, we have to find a code leading to a good tradeoff regarding the performance of the two classes.

5.2. *Finite length simulation results*

In this section, we study the performance of LDPC codes optimized for the AWGN channel with UEP constraints for finite-length codewords and the case of three classes of sensitivity.

The parameters in this section are $\rho(x) = 0.0437x^7 + 0.9563x^8$, $d_{v_{\max}} = 30$, $N = 4096$ and $N = 30000$, $R = 1/2$, $\underline{\alpha} = (0.2, 0.8)$, $\epsilon = 0.1$, and $\epsilon = 0.5$. We compare the BER performance for class 1 and 2 versus E_b/N_0 between the UEP-LDPC code and the AWGN optimized LDPC code after $l = 7$ decoding iterations. As seen in Figures 3 and 4 for BER $= 10^{-5}$ and $\epsilon = 0.1$, we have improved performance for both information classes: about 0.5 dB for class 1 for $N = 4096$ or $N = 30000$, and for the second class, about 0.25 dB for $N = 4096$ and 0.2 dB for $N = 30000$. For $\epsilon = 0.5$, we have improved performance only for the first class: about 0.8 dB for $N = 30000$ and 0.7 dB for $N = 4096$. For the second class,as predicted by asymptotic curves, we have a slight loss in performance: -0.25 dB for $N = 4096$ and -0.2 dB for $N = 30000$.

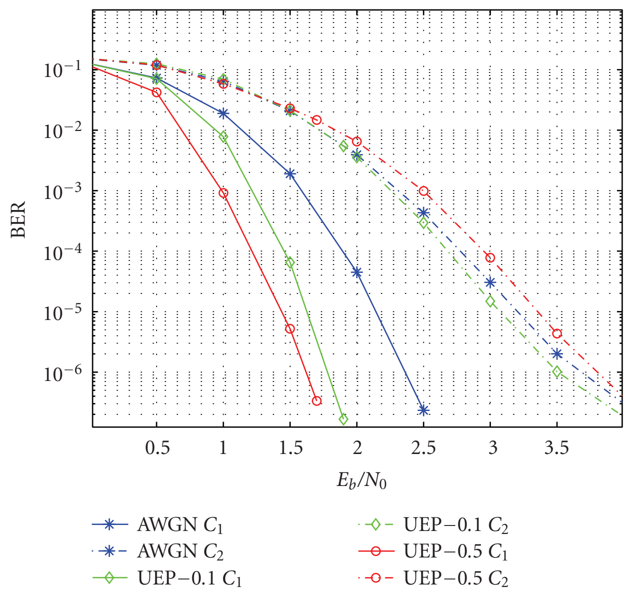

FIGURE 3: Bit error rate performance of different classes versus E_b/N_0 after 7 decoding iterations. We have improved performance for class 1 and 2 for the LDPC code optimized with UEP constraints $N = 4096$.

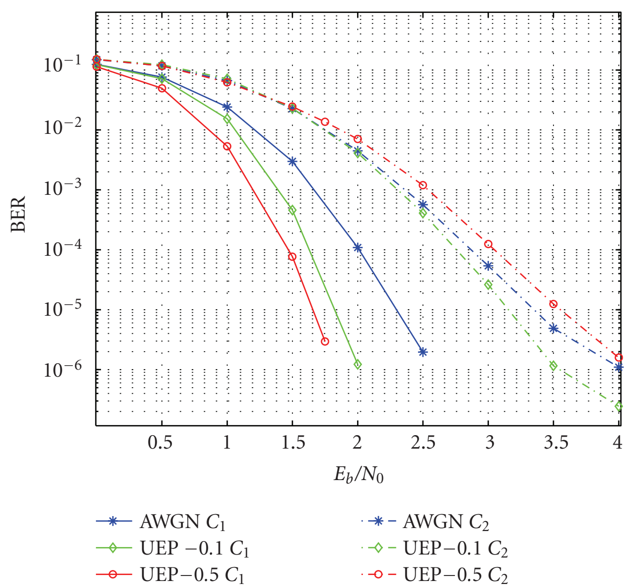

FIGURE 4: Bit error rate performance of different classes versus E_b/N_0 after 7 decoding iterations. We have improved performance for class 1 and 2 for the the LDPC code optimized with UEP constraints $N = 30000$.

6. CONCLUSION

In this paper, we have proposed a general method to optimize LDPC codes under UEP constraints. The proposed strategy takes advantage of the link between UEP and local convergence speed, and is based on the hierarchical optimization of irregularity profiles in each class of sensitivity under specific UEP constraints. As a local objective function, we have proposed the maximization of the average bit node degree in a given class while guaranteeing a minimum degree as high as possible. This strategy shows encouraging results, since the codes optimized with UEP constraints show better UEP capabilities than existing codes. Note that our method could be extended to other types of memoryless channels without difficulties and we successfully applied it in a progressive image transmission context [26].

One should however recall that LDPC codes are very interesting UEP codes if the number of decoding iterations is limited to a reasonable number, which would be fixed by a target latency of the decoder. For example, if the decoder has to work in real time, the number of decoding iterations will be fixed by the channel bit rate as well as the clock frequency and the level of parallelism of the decoder hardware.

ACKNOWLEDGMENTS

This work has partially been presented in IEEE ISIT'04. It was supported by by the French national telecommunication research network (RNRT) project V.I.P.

REFERENCES

[1] J. Hagenauer, "Rate-compatible punctured convolutional (RCPC) codes and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.
[2] A. S. Barbulescu and S. S. Pietrobon, "Rate compatible turbo codes," *Electronics Letters*, vol. 31, no. 7, pp. 535–536, 1995.
[3] W. Henkel and N. Von Deetzen, "Path pruning for unequal error protection turbo codes," in *International Zürich Seminar on Digital Communications*, vol. 2006, pp. 142–145, Zürich, Switzerland,, February 2006.
[4] J. Ha and S. W. McLaughlin, "Optimal puncturing of irregular low-density parity-check codes," in *Proceedings of IEEE International Conference on Communications (ICC '03)*, vol. 5, pp. 3110–3114, Seattle, Wash, USA, May 2003.
[5] C.-F. Lan, K. R. Narayanan, and Z. Xiong, "Scalable image transmission using rate-compatible irregular repeat accumulate (IRA) codes," in *Proceedings of IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 717–720, Rochester, NY, USA, September 2002.
[6] X. Pan, A. H. Banihashemi, and A. Cuhadar, "Progressive transmission of images over fading channels using rate-compatible LDPC codes," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3627–3635, 2006.
[7] C.-F. Lan, Z. Xiong, and K. R. Narayanan, "Source-optimized irregular repeat accumulate codes with inherent unequal error protecticn capabilities and their application to scalable image transmission," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1740–1750, 2006.
[8] C. Poulliat, D. Declercq, C. Lamy-Bergot, and I. Fijalkow, "Analysis and optimization of irregular LDPC codes for joint source-channel decoding," *IEEE Communication Letters*, vol. 9, no. 12, pp. 1064–1066, 2005.
[9] I. M. Boyarinov and G. L. Katsman, "Linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 168–175, 1981.
[10] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 600–607, 1967.

[11] P. S. Guinand, D. Boudreau, and R. Kerr, "Construction of UEP codes suitable for iterative decoding," in *Proceedings of the 6th Canadian Workshop on Information Theory*, pp. 17–20, Kingston, Ontario, Canada, June 1999.

[12] X. Yang, D. Yuan, P. Ma, and M. Jiang, "New research on unequal error protection (UEP) property of irregular LDPC codes," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC '04)*, pp. 361–363, Las Vegas, Nev, USA, January 2004.

[13] M. Luby, M. Mitzenmacher, and A. Shokrollahi, "Pratical loss resilient codes," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pp. 150–159, El Paso, Tex, USA, May 1997.

[14] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.

[15] H. Pishro-Nik, N. Rahnavard, and F. Fekri, "Nonuniform error correction using low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2702–2714, 2005.

[16] N. Rahnavard and F. Fekri, "Unequal error protection using low-density parity-check codes," in *Proceedings of IEEE International Symposium on Information Theory*, p. 449, Chicago, Ill, USA,, June 2004.

[17] N. Rahnavard and F. Fekri, "New results on unequal error protection using LDPC codes," *IEEE Communications Letters*, vol. 10, no. 1, pp. 43–45, 2006.

[18] N. Rahnavard, H. Pishro-Nik, and F. Fekri, "Unequal error protection using partially regular LDPC codes," *IEEE Transactions on Communications*, vol. 55, no. 3, pp. 387–391, 2007.

[19] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[20] X. Lin and W. Wu, "LDPC codes for unequal error protection," in *Proceedings of International Conference on Signal Processing (ICSP '04)*, vol. 2, pp. 1798–1800, Beijing, China, August-September 2004.

[21] V. Kumar and O. Milenkovic, "On unequal error protection LDPC codes based on plotkin-type constructions," *IEEE Transactions on Communications*, vol. 54, no. 6, pp. 994–1005, 2006.

[22] 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.

[23] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, Cambridge, UK, 2005.

[24] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[25] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT Press, Cambridge, Mass, USA, 2000.

[26] C. Poulliat, I. Fijalkow, and D. Declercq, "Scalable image transmission using UEP optimized LDPC codes," in *Proceedings of the 2nd International Symposium on Image/Video Communications over Fixed and Mobile Networks (ISIVC '04)*, pp. 109–112, Brest, France, July 2004.