Comparison of Distributed LDPC Coding Schemes for Decode-and-Forward Relay Channels

Meng Wu*, Petra Weitkemper†, Dirk Wübben*, and Karl-Dirk Kammeyer*
*Department of Communications Engineering, Otto-Hahn-Allee 1, University of Bremen, 28359 Bremen, Germany
Email: {wu, wuebben, kammeyer}@ant.uni-bremen.de
† DoCoMo Communications Laboratories Europe GmbH, Landsberger Straße 312, 80687 München, Germany
Email: weitkemper@docomolab-euro.com

***Abstract*—In order to approach the theoretical limit of the decode-and-forward strategy for the half-duplex relay channel, distributed LDPC coding schemes have been proposed. In these schemes, the code applied at the source should be decodable at the relay to yield correct parity bits. With the help of the parity bits the destination should also be able to estimate the transmitted information correctly. For successful decoding the distributed coding scheme has to be designed jointly, requiring a high design complexity. As an alternative a distributed LDPC scheme based on puncturing is investigated, which requires only the design of one mother code. In this paper we compare three different approaches for designing distributed LDPC codes with respect to their performance and their design complexity.**

I. Introduction

Relay networks are nowadays enjoying increasing interest in wireless communications. With the help of relays, the channel capacity can be enhanced compared to the direct link from the source to the destination. The pioneer studies of the theoretical capacity of relay channels were developed in [1] by Meulen and [2] by Cover and El Gamal considering the Decode-and-Forward (DF) strategy. DF requires the relay to perfectly decode the transmitted codeword, so that the source-relay link dominates the achievable rate of the overall system.

Low-Density Parity-Check (LDPC) codes are known to be powerful due to their capacity-approaching property for single-user communication channels. Therefore, they are considered in this paper to build a relay coding scheme for a half-duplex relay channel using the decode-and-forward strategy. A conventional single-user LDPC code is designed and operates efficiently at a certain channel parameter for the point-to-point transmission from the source to the destination [3]. With the presence of a relay located between the source and the destination using the decode-and-forward strategy, the designed code applied to the source should be successfully decoded at the relay to yield correct additional partiy bits. The parity bits are further transmitted to the destination, forming the overall LDPC code with the code from the source that is decoded at the destination. This requires that the designed constituent LDPC code should operate at two different channel parameters. In other words, the overall LDPC code should be designed jointly for the source-relay link and the source-destination link considering the presence of a relay.

In this paper we make use of the code designs for distributed LDPC coding schemes proposed in [4], [5] and adapt them to our system configuration. In addition, a new scheme that requires much less design complexity is presented, which is based on puncturing proposed in [6], [7] for a point-to-point communication. It is well known that random puncturing degrades the performance of LDPC codes. Therefore this scheme will be sensitive to the parameter settings, especially to the quality of the source-relay link. The performances of the different schemes are compared with each other under Monte-Carlo simulation.

The organization of the paper is as follows. The system model is introduced in Section II. Some basics about LDPC codes and the code design process for a single-user LDPC code based on Density Evolution and Gaussian approximation are presented in Section III. Three distributed LDPC coding schemes, namely, LDPC codes based on a single-user scenario [4], Bi-layer LDPC codes [5] and punctured LDPC codes, are illustrated in details with respect to their code structure and design process in Section IV. The performance and complexity consideration of these three coding schemes are presented and compared to each other in Section V and VI, respectively. The paper is concluded in Section VII.

II. System Model

In this paper, a system with one relay R is considered, which is on the direct line between a source S and a destination D, as shown in Fig. 1.

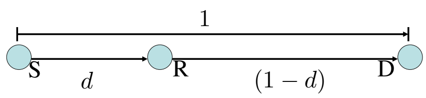


Fig. 1. A relay system with relay R on the direct line between source S and destination D. The distance between the source and the destination is 1 and the distance between the source and relay is d

The relay system operates in a half-duplex mode. Thus, the signal[1] $\mathbf{x}_S$ is transmitted from the source to both the relay and

[1]In this paper, small bold letters represent vectors, capital bold letters represent matrices and italic letters represent variables. For a vector $\mathbf{x}$ of length n, $\mathbf{x}[i]$ is the element in the ith time instant, where $i = 1, 2, .., n$.

This work was supported in part by the German Research Foundation (DFG) under grant Wu 499/7 and Ka 841/20.

the destination in the first time block of length t and the signal $\mathbf{x}_\text{R}$ is transmitted from the relay to the destination while the source keeps silent in the second time block of length $(1-t)$, where t is defined as the time-division factor. The distance between the source and the destination is normalized to 1 and $0 \le d \le 1$ denotes the distance between the source and the relay. By defining the noise terms $\mathbf{n}_\text{SR}$, $\mathbf{n}_\text{RD}$ and $\mathbf{n}_\text{SD}$ for the SR, RD and SD links, respectively, the receive signal $\mathbf{y}_\text{SR}$, $\mathbf{y}_\text{RD}$ and $\mathbf{y}_\text{SD}$ are given by:

$$\mathbf{y}_\text{SR} = \mathbf{x}_\text{S} + \mathbf{n}_\text{SR} \quad (1a)$$

$$\mathbf{y}_\text{SD} = \mathbf{x}_\text{S} + \mathbf{n}_\text{SD} \quad (1b)$$

$$\mathbf{y}_\text{RD} = \mathbf{x}_\text{R} + \mathbf{n}_\text{RD} \quad (1c)$$

The power of each element in both transmit signal $\mathbf{x}_\text{S}$ and $\mathbf{x}_\text{R}$ is normalized to 1. By assuming AWGN channels for all the links with noise powers denoted as σ^2_SR, σ^2_RD and σ^2_SD, respectively, the attenuation or path-loss exponent α is the key concern that builds the relationships between the noise powers of different links. Throughout this paper α is equal to 2 and the following equations hold:

$$\frac{\sigma^2_\text{SR}}{\sigma^2_\text{SD}} = d^\alpha \quad \text{and} \quad \frac{\sigma^2_\text{RD}}{\sigma^2_\text{SD}} = (1-d)^\alpha. \quad (2)$$

Alternatively, the Signal-to-Noise Ratio (SNR) of the different links are given by:

$$\text{SNR}_\text{SD} = \frac{1}{\sigma^2_\text{SD}} \quad (3a)$$

$$\text{SNR}_\text{SR} = \frac{1}{\sigma^2_\text{SR}} = \text{SNR}_\text{SD} \cdot \frac{1}{d^\alpha} \quad (3b)$$

$$\text{SNR}_\text{RD} = \frac{1}{\sigma^2_\text{RD}} = \text{SNR}_\text{SD} \cdot \frac{1}{(1-d)^\alpha} \quad (3c)$$

The *overall relay channel* is defined as the source-destination link including the relay, which contains all the components of the network that form the end-to-end connection. In comparison to a *classical link*, i.e., a system without a relay, the theoretical limit of the decode-and-forward strategy for the overall relay channel is enhanced. Fig. 2 shows for different SNR the capacities for the classical link and for the overall relay channel for BPSK transmission and SR distance d=0.5. Note that the capacity curve for the overall relay channel is shown with respect to the SNR on the SD link, i.e., SNR_SD. The capacity of the overall relay link depends on the capacities of the SD, SR, and RD links and is given by [8]

$$C = \sup_{0\le t\le 1} \min\{t \cdot C_\text{SR}, t \cdot C_\text{SD} + (1-t)\cdot C_\text{RD}\}. \quad (4)$$

It can be observed that the capacity of the overall relay link is 0.4 bits/s/Hz for $\text{SNR}_\text{SD} = -4\,\text{dB}$ as indicated by the cross. In order to achieve the same throughput on a classical link, the required SNR is $\text{SNR}^* = -1.3\,\text{dB}$ as indicated by the circle. This SNR^* is defined as the *effective* SNR and represents the SNR a classical link would need to achieve the same capacity as the relay channel. Obviously, there is a one-to-one relationship between SNR_SD and SNR^* since both curves increase monotonically in Fig. 2.

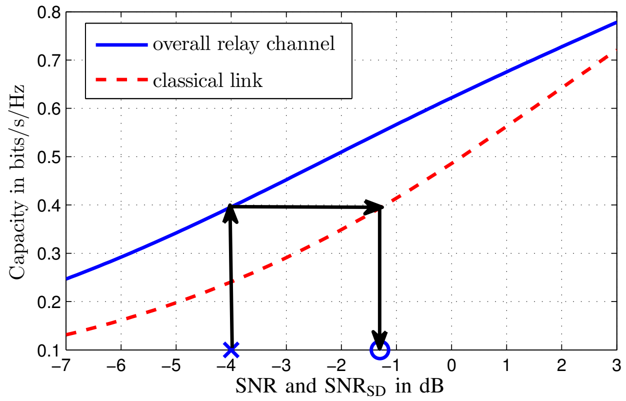


Fig. 2. Capacity vs. SNR for the classical link and the overall relay channel with BPSK modulation. The overall relay channel with SNR_SD can be equivalently treated as a classical link with SNR^*.

The overall task is to design a distributed LDPC code which is decodable at the relay and the destination. To this end, the effective SNR SNR^* is used for the code design.

III. Introduction to LDPC Codes

A. LDPC Codes defined by the Parity-Check Matrix

LDPC codes are a special type of linear block codes, with the specialties that their codeword length n is usually quite large and their parity-check matrix $\mathbf{H}$ of size $n \times (n-k)$ is always sparse. The information vector $\mathbf{u}$ of length k is encoded by the $k \times n$ generator matrix $\mathbf{G}$ to yield the codeword $\mathbf{b} = \mathbf{u} \otimes \mathbf{G}$. $\mathbf{b}$ is a $1 \times n$ vector and $\mathbf{G} \otimes \mathbf{H}^T = \mathbf{0}$ holds. Additionally, $\mathbf{u}$, $\mathbf{b}$, $\mathbf{G}$ and $\mathbf{H}$ are all in the finite field GF(2). By definition, each bit of the codeword or each column of $\mathbf{H}$ represents a variable node denoted as v, and each constraint or each row of $\mathbf{H}$ represents a check node denoted as c. There is an edge between a variable node and a check node if and only if their intersect in $\mathbf{H}$ is 1. Every parity-check matrix can be represented graphically by a factor graph [9]. As an example, a parity-check matrix $\mathbf{H}$ and its corresponding factor graph are shown in Fig. 3, where the variable nodes v_i, $i = 0, 1, 2, 3$ are represented by circles and the check nodes c_j, $j = 0, 1, 2$ are represented by squares.

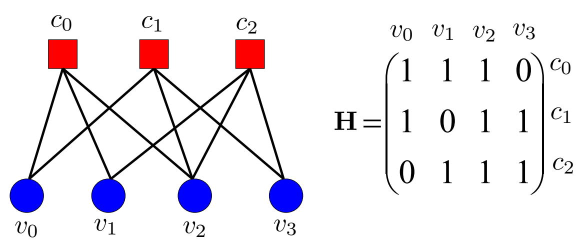


Fig. 3. An example for a parity-check matrix $\mathbf{H}$ and its factor graph

A degree-i variable node is connected to i edges and a degree-j check node is connected to j edges. The proportion of edges connected to a degree-i variable node over the total number of edges is denoted as λ^E_i and the proportion of edges connected to a degree-j check node over the total number of edges is defined as ρ_j^E. To describe the edge connections in a factor graph, the degree distribution for variable nodes on an edge perspective is defined by the polynomial [10]

$$\lambda^E(w) = \lambda_2^E w + \lambda_3^E w^2 + ... + \lambda_{d_v}^E w^{d_v-1} = \sum_{i=2}^{d_v} \lambda_i^E w^{i-1}. \quad (5)$$

Similarly, the degree distribution for check nodes on an edge perspective is

$$\rho^E(w) = \rho_2^E w + \rho_3^E w^2 + ... + \rho_{d_c}^E w^{d_c-1} = \sum_{j=2}^{d_c} \rho_j^E w^{j-1}. \quad (6)$$

The parameters d_v and d_c represent the maximum values of variable and check degrees, respectively. Additionally, both λ_1^E and ρ_1^E equal to 0 since a node connected with only one edge is not allowed. We also define the proportion of degree-i variable nodes over the total number of variable nodes as λ_i^N and the proportion of degree-j check nodes over the total number of variable nodes as ρ_j^N. This yields the degree ditributions on a node perspective [10]

$$\lambda^N(w) = \sum_{i=2}^{d_v} \lambda_i^N w^i \quad \text{and} \quad \rho^N(w) = \sum_{j=2}^{d_c} \rho_j^N w^j. \quad (7)$$

The degree distributions on an edge perspective in (5), (6) and on a node perspective in (7) are essentially identical since they are connected by the relation [10]

$$\lambda_i^N = \frac{\lambda_i^E/i}{\int_0^1 \lambda^E(x)dx} \quad \text{and} \quad \rho_j^N = \frac{\rho_j^E/j}{\int_0^1 \rho^E(x)dx}. \quad (8)$$

The code rate R_c can be calculated using the degree distributions (5), (6) or, alternatively, using (7) based on (8) [10]

$$R_c = 1 - \frac{\int_0^1 \rho^E(w)dw}{\int_0^1 \lambda^E(w)dw} = 1 - \frac{\sum_{j=2}^{d_u} \rho_j^E/j}{\sum_{i=2}^{d_v} \lambda_i^E/i} = 1 - \frac{\sum_{i=2}^{d_v} i\,\lambda_i^N}{\sum_{j=2}^{d_c} j\,\rho_j^N}. \quad (9)$$

The degree distributions $\lambda^E(w)$ and $\rho^E(w)$, or $\lambda^N(w)$ and $\rho^N(w)$, fully depict an LDPC ensamble. Therefore, the design of LDPC codes is to optimize these degree distributions, as shown in the following subsection.

B. Message Passing based on Factor Graph for LDPC Codes

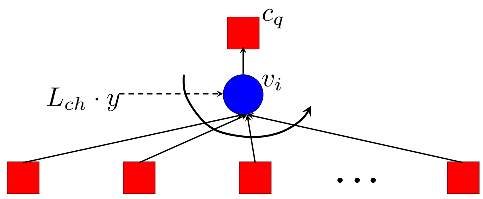
Density Evolution has been applied to design LDPC codes by keeping track of soft messages [11] flowing between variable nodes and check nodes. We define $\mathcal{M}_i$ as the set of all check nodes connected to the variable node v_i and $L_{ch} \cdot y$ as the LLR value from the channel. L_{ch} is the reliability of the channel and equal to $\frac{2}{\sigma^2}$, with σ^2 the variance of the channel noise, and y is the receive signal. The updating rule of soft messages for the variable node v_i whose outgoing message will flow to a check node c_q is

$$L\left(v_i^q\right) = \sum_{\substack{j \in \mathcal{M}_i \\ j \neq q}} L\left(c_j\right) + L_{ch} \cdot y \quad (10)$$

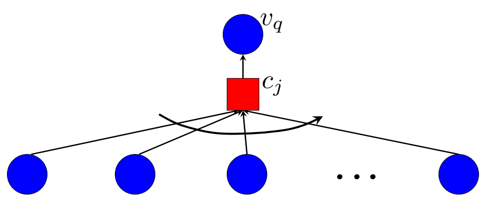
where $L()$ is the function that calculates the LLR value of a certain node, and represents the probability for a correct decision. Thus, the LLR $L\left(v_i^q\right)$ of the variable node v_i is given by the sum of the channel information $L_{ch} \cdot y$ and all LLR values $L\left(c_j\right)$, $j \in \mathcal{M}_i$, except the check node c_q the outgoing message will flow to, as illustrated in Fig. 4(a). Similarly, by defining $\mathcal{K}_j$ as the set of all variable nodes connected to a check node c_j, the updating rule of soft messages for the check node c_j is

$$\tilde{c}_j^q = \tanh\frac{L\left(c_j\right)}{2} = \prod_{\substack{i \in \mathcal{K}_j \\ i \neq q}} \tanh\frac{L\left(v_i\right)}{2} \quad (11)$$

when the output message will be fed to a variable node v_q. Function $\tanh\frac{L()}{2}$ is the expectation of the decision and is called soft bit of a certain node [11]. Therefore, the soft bit $\tilde{c}_j$ of the check node c_j is given by the product of all soft bits $\tilde{v}_i = \tanh\frac{L(v_i)}{2}$, $i \in \mathcal{K}_j$ other than the variable node v_q, as illustrated in Fig. 4(b). Both (10) and (11) are based on the Sum-Product algorithm [9].



(a) The updating rule of a variable node v_i when the output flows to the check node c_q: the messages are collected from the channel and all check nodes in $\mathcal{M}_i \backslash q$



(b) The updating rule of a check node c_j when the output flows to the variable node v_q: the messages are collected from all variable nodes in $\mathcal{K}_j \backslash q$

Fig. 4. The message passing process for the code design of LDPC codes

C. Design of a Single-user LDPC Code using Density Evolution based on Gaussian Approximation

Density Evolution requires to keep track of the whole density of the soft messages $L\left(v_i\right)$ and $\tilde{c}_j$, actually resulting in a numerically unsolvable problem [10]. To simplify the density-tracking process, all soft message densities are assumed to be symmetric Gaussian distributed. A symmetric Gaussian distribution $g(w)$ is a Gaussian distribution with the property $g(w) = g(-w)e^w$. It can be completely defined by its mean m_g since its variance σ_g^2 is twice of its mean, i.e., $\sigma_g^2 = 2m_g$ [12]. Therefore, instead of keeping track of the whole density of soft messages, only the mean has to be kept track of in Density Evolution, reducing the infinite-dimensional problem to a one-dimensional problem. We use the Density Evolution based on Gaussian approximation to optimize the degree distributions of LDPC codes for a given initial value L_{ch}, where L_{ch} is the mean of the LLR from the channel. For brevity, only the procedure of the code design process is presented without the steps in its derivation, which can be found in [12]. The design problem can be formulated as an optimization problem to maximize the code rate R_c defined in (9):

$$\text{max. } 1 - \frac{\sum_{j=2}^{d_c} \rho_j^E / j}{\sum_{i=2}^{d_v} \lambda_i^E / i} \tag{12a}$$

$$\text{s.t. } \sum_{i=2}^{d_v} \lambda_i^E = 1 \tag{12b}$$

$$\sum_{i=2}^{d_v} \lambda_i^E \left(f_i(L_{ch}, p) - p \right) < 0 \tag{12c}$$

$$\lambda_2^E - \frac{e^{1/2\sigma^2}}{\sum_{j=2}^{d_r} \rho_j^E (j-1)} < 0 \,. \tag{12d}$$

Throughout this paper, the degree distribution for check nodes $\rho^E(w)$ is pre-fixed since it is hard to optimize $\lambda^E(w)$ and $\rho^E(w)$ simultaneously. Therefore, relation (12a) is equivalent to maximizing $\sum_{i=2}^{d_v} \lambda_i^E / i$. The completeness condition (12b) ensures that the sum of all element λ_i^E in the degree distribution $\lambda^E(w)$ is 1. The successful decoding condition is fulfilled by (12c) in the limit of infinite codeword length and iterations, where p decreases from 1 to 0 if the decoding succeeds. The functions $f_i(L_{ch}, p) - p, i = 2, 3.., d_v$ are the set of elementary EXIT charts and their linear combination $\sum_{i=2}^{d_v} \lambda_i^E f_i(L_{ch}, p) - p$ is the overall EXIT chart denoted as $f(L_{ch}, p)$ [12]. The stability condition (12d) ensures that the EXIT charts converge as the iteration grows to infinity, with more details in [4], [12]. Noticeably, (12a)-(12d) are all linear combinations of the degree distribution λ_i^E to be optimized. Thus the optimization problem can be solved by linear programming [13].

IV. Distributing LDPC Codes to Relay Channels

A. General LDPC Code Distributing Strategy

To distribute an LDPC code to the relay system in Fig. 1, the information vector $\mathbf{u}$ at the source is first encoded by multiplication with the generator matrix $\mathbf{G}_1$ to yield the codeword $\mathbf{b}_\text{S} = \mathbf{u} \otimes \mathbf{G}_1$. The corresponding parity-check matrix is denoted as $\mathbf{H}_1$, where $\mathbf{G}_1 \otimes \mathbf{H}_1^T = \mathbf{0}$ holds. Considering BPSK modulation, the signal $\mathbf{x}_\text{S} = \mathbf{1} - \mathbf{2} \cdot \mathbf{b}_\text{S}$ is transmitted to the relay and the destination simultaneously in the first time block. The destination stores the received message $\mathbf{y}_\text{SD} = \mathbf{x}_\text{S} + \mathbf{n}_\text{SD}$ for further use. The relay, after receiving the message $\mathbf{y}_\text{SR} = \mathbf{x}_\text{S} + \mathbf{n}_\text{SR}$, performs LDPC decoding for the source code using $\mathbf{H}_1$ to recover the original message. The estimated codeword $\hat{\mathbf{b}}_\text{S}$ is further *compressed* by multiplication with another parity-check matrix $\mathbf{H}_2$, i.e., $\mathbf{b}_\text{R} = \hat{\mathbf{b}}_\text{S} \otimes \mathbf{H}_2^T$. The *compressed* bits $\mathbf{b}_\text{R}$, called side information or syndrome, are BPSK-modulated to yield the signal $\mathbf{x}_\text{R} = \mathbf{1} - \mathbf{2} \cdot \mathbf{b}_\text{R}$, which is transmitted to the destination without any further code protection in the second time block, i.e., $\mathbf{y}_\text{RD} = \mathbf{x}_\text{R} + \mathbf{n}_\text{RD}$. Note that $\mathbf{b}_\text{R}$ acts as extra parity bits that will aid the decoding at the destination. Finally, $\mathbf{y}_\text{SD}$ and $\mathbf{y}_\text{RD}$ are collected at the destination to perform LDPC decoding for the overall code using the extended form $\mathbf{H}_\text{ext}$ of $\mathbf{H}$, where

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_\text{ext} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{H}_2 & \mathbf{I} \end{bmatrix}. \tag{13}$$

Assuming error free transmission on all channels, the signal transmitted by the relay corresponds to $\mathbf{b}_\text{R} = \mathbf{b}_S \otimes \mathbf{H}_2^T$ and the check equation at the destination (ignoring BPSK modulation) reads

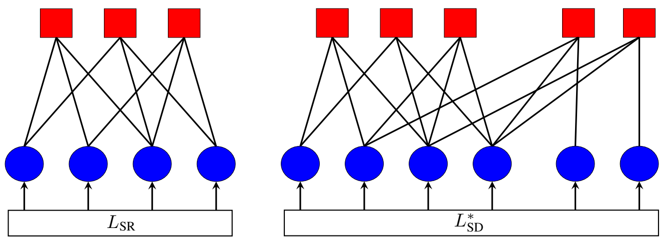
$$[\mathbf{b}_\text{S} \;\; \mathbf{b}_\text{R}] \otimes \mathbf{H}_\text{ext}^T = [\mathbf{b}_\text{S} \;\; \mathbf{b}_\text{R}] \otimes \begin{bmatrix} \mathbf{H}_1^T & \mathbf{H}_2^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{14a}$$

$$= \left[\mathbf{b}_\text{S} \otimes \mathbf{H}_1^T \;\; \mathbf{b}_\text{S} \otimes \mathbf{H}_2^T \oplus \mathbf{b}_\text{R} \right] \,. \tag{14b}$$

The term $\mathbf{b}_\text{S} \otimes \mathbf{H}_1^T = \mathbf{u} \otimes \mathbf{G}_1 \otimes \mathbf{H}_1^T$ is zero due to $\mathbf{G}_1 \otimes \mathbf{H}_1^T$. The second term should also be zero, as the signal transmitted by the relay was chosen to be $\mathbf{b}_\text{R} = \mathbf{b}_\text{S} \otimes \mathbf{H}_2^T$.

In order to design the parity check matrices $\mathbf{H}_1$ and $\mathbf{H}$ jointly such that $\mathbf{H}_1$ works for the SR link and $\mathbf{H}$ works for the overall relay channel, two schemes have been proposed, known as LDPC codes based on a single-user scenario [4] and Bi-layer LDPC codes [5]. In both [4] and [5], the extra parity bits $\mathbf{b}_\text{R}$ from the relay are re-encoded by another strong LDPC codebook so that they are assumed to be perfectly known at the destination, whereas $\mathbf{b}_\text{R}$ is uncoded in this paper. In order to decrease the design complexity caused by the joint optimization of two codes, we propose a simpler scheme based on puncturing in Subsection IV-D. All the three schemes are studied in the sequel.

B. LDPC Codes based on a Single-user Scenario



(a) The source code is decoded at the relay using $\mathbf{H}_1$, where the variable nodes receive L_SR.

(b) The overall code is decoded at the destination using $\mathbf{H}$, where the variable nodes receive L^*_SD. The overall relay channel is treated as a direct link.

Fig. 5. Factor graph of the source code and overall code for the code optimization of LDPC codes based on a single-user scenario

To jointly design the source code and the overall code, both $\mathbf{H}_1$ and $\mathbf{H}$ are treated as single-user LDPC codes in this scheme [4]. As shown in Fig. 5(a) for the optimization of $\mathbf{H}_1$, all the variable nodes of $\mathbf{H}_1$ receive the mean $L_\text{SR} = \frac{2}{\sigma_\text{SR}^2} = 2\,\text{SNR}_\text{SR}$ of the LLR from the SR link, and follows the code design method presented in Subsection III-C for a single-user LDPC code. To optimize $\mathbf{H}$, some extra parity bits $\mathbf{x}_\text{R}$ are transmitted from the relay to the destination, making the overall factor graph $\mathbf{H}$ extend as shown in Fig. 5(b). $\mathbf{H}$ is treated as another single-user LDPC code, and all its variable nodes receive the same mean $L_{\text{SD}}^{*} = \frac{2}{\sigma_{\text{SD}}^{2*}} = 2\,\text{SNR}_{\text{SD}}^{*}$ from the overall relay channel, and still follows the code design method for a single-user LDPC code in Subsection III-C.

The degree distributions of $\mathbf{H}_1$ are denoted as $\lambda_1^N(w)$ and $\rho_1^N(w)$, and the degree distributions of $\mathbf{H}$ are denoted as $\lambda^N(w)$ and $\rho^N(w)$, all of which are on a node perspective, as in (7). Degree distributions on a node perspective are in practice for the joint code design based on a single-user scenario because (15d) has to be expressed on a node perspective to form a linear combination, as shown in the sequel. The joint design problem can now be viewed as a larger linear programming problem to maximize the code rate R_c of the overall relay channel stated as follows:

$$\text{max. } 1 - \frac{\sum_{i=2}^{d_v} i\,\lambda_i^N}{\sum_{j=2}^{d_c} j\,\rho_j^N} \tag{15a}$$

$$\text{s.t. } 1 - \frac{\sum_{i=2}^{d_{v,1}} i\,\lambda_{1,i}^N}{\sum_{j=2}^{d_{c,1}} j\,\rho_{1,j}^N} = R_{c,\text{S}} \tag{15b}$$

$$\sum_{i=2}^{d_{v,1}} \lambda_{1,i}^N = 1, \quad \sum_{i=2}^{d_v} \lambda_i^N = 1 \tag{15c}$$

$$\sum_{i=j}^{d_v} \left(\lambda_{1,i}^N - \lambda_i^N\right) \leq 0, \quad \forall j = 2, 3, ..., d_v \tag{15d}$$

$$\sum_{i=2}^{d_{v,1}} i\,\lambda_{1,i}^N \left(f_i(L_{\text{SR}}, p) - p\right) < 0 \tag{15e}$$

$$\sum_{i=2}^{d_v} i\,\lambda_i^N \left(f_i(L_{\text{SD}}^{*}, p) - p\right) < 0 \tag{15f}$$

$$2\lambda_{1,2}^N - \frac{e^{1/2\sigma_{n,\text{SR}}^2}}{\sum_{j=2}^{d_{c,1}} \rho_{1,j}(j-1)} \sum_{i=2}^{d_{v,1}} i\,\lambda_{1,i}^N < 0 \tag{15g}$$

$$2\lambda_2^N - \frac{e^{1/2\sigma_{n,\text{SD}}^{2*}}}{\sum_{j=2}^{d_c} \rho_j(j-1)} \sum_{i=2}^{d_v} i\,\lambda_i^N < 0\,. \tag{15h}$$

In (15), $d_{v,1}$ and d_v represent the maximum values of variable degrees for $\mathbf{H}_1$ and $\mathbf{H}$, respectively, and $d_{c,1}$ and d_c represent the maximum values of check degrees for $\mathbf{H}_1$ and $\mathbf{H}$, respectively. Note that in this optimization we fix the source code rate $R_{c,\text{S}}$ and maximize that of the overall relay channel R_c since it's not possible to maximize two targeting terms simutaneously in one optimization. The fixed code rate $R_{c,\text{SR}}$ works as an equality constraint of the whole linear programming problem (see (15b)). The completeness condition for the degree ditributions are fulfilled by (15c). Additionally, $\mathbf{H}_1$ is always a subgraph of $\mathbf{H}$, which is guaranteed by (15d). Similarly to (12c) for a single-user LDPC code, the successful decoding conditions for the SR link and the overall relay channel are presented by (15e) and (15f), respectively. Finally, the stability conditions are fulfilled by (15g) and (15h), similarly to (12d).

C. Bi-layer LDPC Codes

Comparing with LDPC codes based on a single-user scenario, Bi-layer LDPC codes treat the check nodes in $\mathbf{H}_1$ and $\mathbf{H}_2$ differently, which are defined as 'left checks c_L' and 'right checks c_R', respectively. This requires different updating rules for the left check nodes and right check nodes based on the two-dimensional degree distribution. A degree-(i,k) variable node v is defined as a variable node that is connected with i edges in $\mathbf{H}_1$ and k edges in $\mathbf{H}_2$. Then the proportion of edges connected to a degree-(i,k) variable node over the total number of edges is denoted as $\lambda_{i,k}$. With the help of this notation, the two-dimensional degree distribution for variable nodes on an edge perspective is defined as

$$\lambda^E(w,z) = \sum_{i=2}^{d_{v,1}} \sum_{k=0}^{d_{v,2}} \lambda_{i,k}^E w^{i-1} z^{k-1}, \tag{16}$$

where $d_{v,1}$ and $d_{v,2}$ represent the maximum values of variable degrees in $\mathbf{H}_1$ and $\mathbf{H}_2$, respectively. Note that k starts at 0 since variable nodes that have no connections with $\mathbf{H}_2$ are allowed. This is slightly different from the degree distribution defined in (5), where λ_0^E and λ_1^E are zero.

To jointly design $\mathbf{H}_1$ and $\mathbf{H}$, $\mathbf{H}_1$ is still treated as a single-user LDPC code, whose variable nodes receive the mean L_{SR} from the SR link, as shown in Fig. 6(a), and follows the code design method presented in Sec. III-C. For the overall code decoded at the destination, the bi-layer density evolution is briefly introduced, more details can be found in [5]. For a variable node v_i in $\mathbf{H}$, $\mathcal{M}_{L,i}$ is defined as the set of left check nodes c_L connected to v_i and $\mathcal{M}_{R,i}$ is defined as the set of right check nodes c_R connected to v_i. Similarly to (10), the updating rule at the variable node v_i is

$$L\left(v_i^q\right) = \sum_{\substack{\ell \in \mathcal{M}_{L,i} \\ \ell \neq q}} L\left(c_{L,\ell}\right) + \sum_{\ell \in \mathcal{M}_{R,i}} L\left(c_{R,\ell}\right) + L_{\text{SD}} \cdot y_{\text{SD}} \tag{17}$$

if the outgoing message will flow to a left check node $c_{L,q}$. Similarly, if the outgoing message will be fed to a right check node $c_{R,q}$, the updating rule at the variable node v_i is

$$L\left(v_i^q\right) = \sum_{\ell \in \mathcal{M}_{L,i}} L\left(c_{L,\ell}\right) + \sum_{\substack{\ell \in \mathcal{M}_{R,i} \\ \ell \neq q}} L\left(c_{R,\ell}\right) + L_{\text{SD}} \cdot y_{\text{SD}} \tag{18}$$

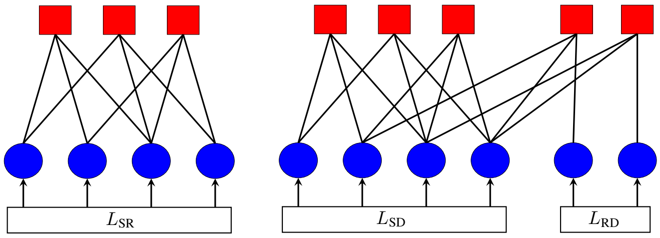
where $L_{\text{SD}} = \frac{2}{\sigma_{\text{SD}}^2} = 2\,\text{SNR}_{\text{SD}}$ is the LLR from the direct SD link. For a left check node $c_{L,j}$, $\mathcal{K}_{L,j}$ is defined as the set of variable nodes in $\mathbf{H}$ connected to $c_{L,j}$. Similarly to (11), the updating rule at the left check node $c_{L,j}$ is

$$\tilde{c}_{L,j}^q = \tanh\frac{L\left(c_{L,j}\right)}{2} = \prod_{\substack{l \in \mathcal{K}_{L,j} \\ l \neq q}} \tanh\frac{L\left(v_l\right)}{2} \tag{19}$$

if the destination of the outgoing message is a variable node v_q. The set of variable nodes in $\mathbf{H}$ connected to $c_{R,j}$ is denoted as $\mathcal{K}_{R,j}$ for a right check node $c_{R,j}$. Subsequently, if the outgoing message will be passed to a variable node v_q, the updating rule at the right check node $c_{R,j}$ is

$$\tilde{c}^{q}_{R,j} = \tanh\frac{L\left(c_{R,j}\right)}{2} = \tanh\frac{L_{\text{RD}}\cdot y_{\text{RD}}}{2}\cdot\prod_{\substack{l\in\mathcal{K}_{R,j}\\ l\neq q}}\tanh\frac{L\left(v_l\right)}{2} \tag{20}$$

where $L_{\text{RD}} = \frac{2}{\sigma^2_{\text{RD}}} = 2\,\text{SNR}_{\text{RD}}$ is the channel reliability from the RD link. Note that in contrast to the left check nodes, the right check nodes also receive the LLR $L_{\text{RD}}\cdot y_{\text{RD}}$ from the RD link, as can be observed in (20) and Fig. 6(b).



(a) The source code is decoded at the relay using $\mathbf{H}_1$, where the variable nodes receive L_{SR}.

(b) The overall code is decoded at the destination using $\mathbf{H}$, where the variable nodes in the first time block receive L_{SD} and those in the second time block receive L_{RD}.

Fig. 6. Factor graph of the source code and overall code for the code optimization of Bi-layer LDPC codes

The joint design problem can now be viewed again as a linear programming problem stated as follows [5]

$$\text{max.}\ \sum_{i\geq 2}\sum_{k\geq 0}\frac{1}{i+k}\lambda^{E}_{i,k} \tag{21a}$$

$$\text{s.t.}\ \sum_{i\geq 2}\sum_{k\geq 0}\lambda^{E}_{i,k} = 1 \tag{21b}$$

$$\sum_{i\geq 2}\sum_{k\geq 0}\frac{i}{i+k}\lambda^{E}_{i,k} = \eta \tag{21c}$$

$$\sum_{i\geq 2}\sum_{k\geq 0}\frac{i}{i+k}\lambda^{E}_{i,k} f_i(L_{\text{SR}}, p) < \eta p \tag{21d}$$

$$\sum_{i\geq 2}\sum_{k\geq 0}\lambda^{E}_{i,k}\left(\frac{i}{i+k}f^{L}_{i,k}\left(L_{\text{SD}}, L_{\text{RD}}, p^L, p^R\right) + \frac{k}{i+k}f^{R}_{i,k}\left(L_{\text{SD}}, L_{\text{RD}}, p^L, p^R\right)\right) < \eta p^L + (1-\eta)\,p^R \tag{21e}$$

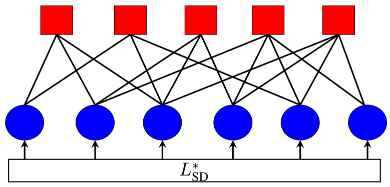
$$\sum_{j\geq 0}\frac{2}{2+j}\lambda^{E}_{2,j} - \frac{e^{1/2\sigma^2_{\text{SR}}}}{\sum_{j=2}^{d_{c,1}}\rho^{E}_{1,j}(j-1)} < 0 \tag{21f}$$

$$\lambda^{E}_{2,0} - \frac{e^{1/2\sigma^2_{\text{SD}}}}{\sum_{j=2}^{d_c}\rho^{E}_{j}(j-1)} < 0\,. \tag{21g}$$

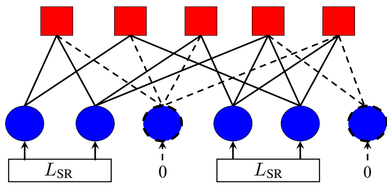
In the optimization of Bi-layer LDPC codes, the code rates $R_{\text{c,S}}$ of the source code and R_c of the overall code are jointly maximized by (21a). The completeness condition for the degree distribution is fulfilled by (21b). Additionally, η is defined as the proportion of edges belonging to $\mathbf{H}_1$ over the total number of edges in $\mathbf{H}$. η can be calculated by the two-dimensional degree distribution $\lambda^E(w,z)$ as shown in (21c), and works as an equality constraint for the optimization. (21d) is the successful decoding condition for the SR link. (21e) is derived from (17)-(20) and represents the successful decoding condition for the overall relay channel, where $\left(p^L, p^R\right)$ decreases from $(1,1)$ to $(0,0)$ as iterations go on. The stability conditions are fulfilled by (21f) and (21g).

D. Punctured LDPC Codes

The design process of both the previous schemes calls for joint optimization of two LDPC codes based on density evolution. To avoid the complicated derivation and calculation, we propose a much simpler scheme to design distributed LDPC codes using puncturing, as done, e.g., in [14] for convolutional codes. First, a mother code $\mathbf{H}$ with code rate R_c is designed either by linear programming as addressed in Sec. III-C or by using the codes from Urbanke's website [15] as a single-user LDPC code to suit the channel condition of the overall relay channel. Then a proportion of parity bits are punctured out randomly in order to meet the channel condition of the SR link. The code rate is raised to $R_{\text{c,S}}$ due to puncturing. Note that there exist other sophisticated puncturing patterns that achieve better performances [6], [7], but only random puncturing is considered in this paper.



(a) The mother code is decoded at the destination using $\mathbf{H}$, where all the variable nodes receive L^*_{SD}, as the overall relay channel is treated equivalently as a direct link.



(b) The punctured code is decoded at the relay still using $\mathbf{H}$, where the un-punctured variable nodes receive L_{SR} and the punctured nodes (- -) receive 0 LLR value.

Fig. 7. Factor graph of the source code and overall code for the code optimization of punctured LDPC codes

Fig. 7 illustrates the process above. The whole graph is designed by taking L^*_{SD} from the overall relay channel. Then some bits are punctured out, which are represented by the dashed parts in Fig. 7(b). The un-punctured variable nodes receive L_{SR} from the SR link while the punctured nodes receive zero LLR values and get recovered as the decoding iteration at the relay goes on. The bits punctured at the source are now added by the relay, and the destination receives the whole codeword, but with different SNRs, i.e., SNR_{SR} for the source code and SNR_{RD} for the additional parity bits. Note that the simplicity of this scheme locates on the fact that the design only depends on the channel condition of the overall relay channel, and just the puncturing probability is adapted to the certain relay position. This is an indicator that there is no complex joint optimization of two codes. Furthermore, the same decoder $\mathbf{H}$ is used at both the relay and the destination for punctured LDPC codes.

V. Simulation Results

To compare the performances of the three distributed coding schemes, their Bit Error Rate (BER) at the relay for the SR link and at the destination for the overall relay channel are simulated with codeword length of $n = 150.000$. For the simulations 100 decoding iterations have been carried out at the relay and at the destination.

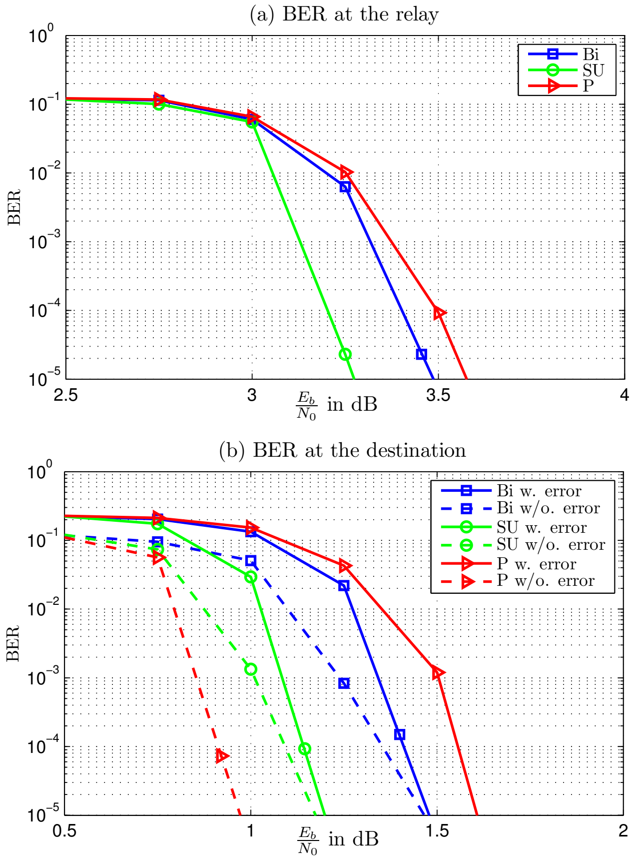


Fig. 8. BER performance of three distributed LDPC coding schemes for $d = 0.5$ with BPSK modulation. The code rates are $R_{c,S} = 0.8$ and $R_c = 0.55$ for all three schemes.

Fig. 8 shows the performance of the three distributed LDPC coding schemes discussed in this paper for the SR distance $d = 0.5$. After decoding the source code at the relay, Bi-layer LDPC codes (Bi) outperform Punctured LDPC codes (P) while LDPC codes based on a single-user scenario (SU) achieve the best performance, as shown in Fig. 8(a). The BER at the destination under the assumption of error-free decoding at the relay are shown in Fig. 8(b). In this case, punctured LDPC codes perform the best, as the overall LDPC code is optimized for this case. However, when error propagation is taken into account under the realistic assumption that the relay may not be able to decode correctly, the overall performance of the punctured code is significantly degraded due to puncturing, which increases the code rate from 0.55 to 0.8. The performances of the other two codes are not much changed, as the source code decoded at the relay was considered during the design process. This is an indicator that the overall performance strongly depends on the robustness of the SR link. With error propagation considered, LDPC codes based on a single-user scenario achieve the best performance while punctured LDPC codes achieve the worst.

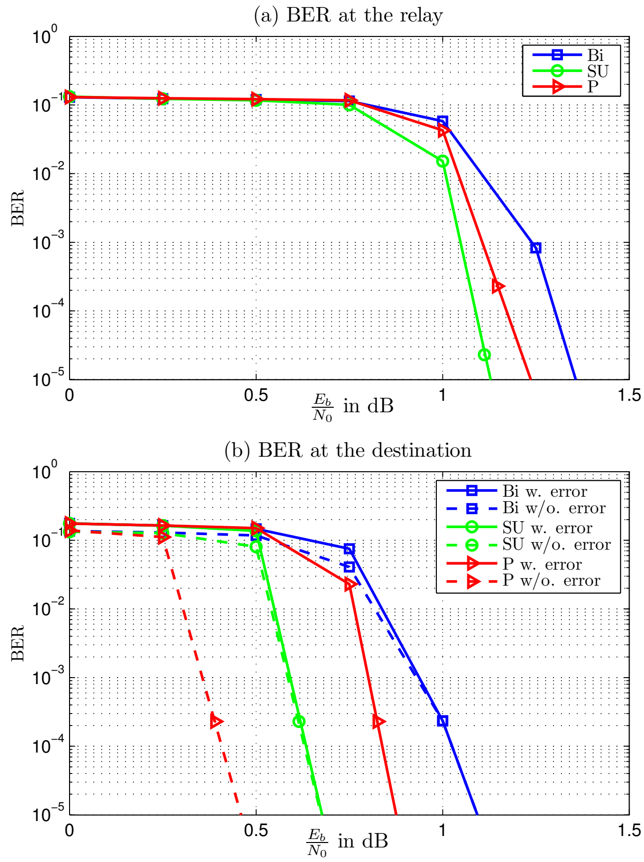


Fig. 9. BER performance of three distributed LDPC coding schemes for $d = 0.8$ with BPSK modulation. The code rates are $R_{c,S} = 0.6$ and $R_c = 0.5$ for all three schemes.

Fig. 9 shows the performance of the three schemes again when the SR distance $d = 0.8$. The dependency of the overall relay channel on the SR link is again visualized. It is shown that in this case, with smaller amount of punctured bits, punctured LDPC codes outperform Bi-layer LDPC codes while LDPC codes based on a single-user scenario still work well and achieve the best performance. This means, that punctured LDPC codes, which claim a much simpler code design strategy, result in a more reliable SR link when less bits are punctured out. The code rate in this case is increased from 0.5 to 0.6 due to puncturing.

Additionally, by comparing Fig. 8(b) and Fig. 9(b), it can be observed that error propagation causes greater performance degradation when the relay is closer to the source for all the three coding schemes. This is because more extra parity bits are generated at the relay and transmitted to the destination in this case, and the decoding performance at the destination depends more heavily on the extra information from the relay.

VI. Complexity Consideration

First, the design complexity of LDPC codes based on a single-user scenario and Bi-layer LDPC codes are compared. Obviously, LDPC codes based on a single-user scenario treat all check nodes uniformly while Bi-layer LDPC codes distinguish left and right check nodes, indicating Bi-layer LDPC codes should be more precise but require higher complexity. Because of the presence of different check nodes, the elementary EXIT charts for Bi-layer LDPC codes have two dimensions p^L and p^R as defined in (21e), whereas the elementary EXIT charts for single-user LDPC codes only have one dimension p as defined in (12c). To deal with the continuous elementary EXIT charts with computers for both approaches, they have to be discretized. We assume the parameter p to be represented by, e.g., 400 samples. This would lead to 400 constraints for a single-user LDPC code, but to $400^2 = 160000$ constraints for a Bi-layer LDPC code because two parameters p^L and p^R are present. So, either the complexity has to be increased dramatically for the Bi-layer code or the number of samples has to be decreased. However, when the overall number of constraints is limited to 400, as confined by the funtion $linprog()$ in MATLAB, the number of samples per dimension is only 20, leading to the bad performance of the Bi-layer LDPC codes shown in the previous section.

Secondly, it is quite clear that punctured LDPC codes require much less design complexity compared with the other two schemes. Both of LDPC codes based on a single-user scenario and Bi-layer LDPC codes call for joint optimization of two codes that are strongly connected (one being the subgraph of the other). Different decoders are also needed at the relay and the destination. For punctured LDPC codes, only one standard single-user LDPC code has to be optimized as a mother code. Subsequently, some bits are punctured out to form the source code. The same decoder can also be used at both the relay and the destination. Furthermore, both LDPC codes based on a single-user scenario and Bi-layer LDPC codes have to be designed for each triple of SNRs, namely, SNR_{SR}, SNR_{SD} and SNR_{RD}, related to a relay network, whereas the punctured LDPC codes just have to be optimized for the overall relay channel. Therefore, much fewer codes have to be designed for the same amount of scenarios. This leads to much lower complexity.

VII. Conclusion

In this paper, two distributed LDPC coding schemes proposed in the literature, namely, LDPC codes based on a single-user scenario and Bi-layer LDPC codes, have been modified and adapted for the half-duplex decode-and-forward relay channel with orthogonal access of the source and the relay. Based on Gaussian approximation for Density Evolution, the code design process for both the two schemes can be summarized as a linear programming problem. However, due to their joint optimization characteristic, both of the schemes require quite a high design complexity. Inspired by the simple idea of puncturing, a much simpler scheme to design the distributed LDPC codes, has been proposed and is called punctured LDPC codes.

Comparisons of the three schemes with respect to their performance show, that punctured LDPC codes are superior in comparison with Bi-layer LDPC codes when the relay is near the destination and vice versa. LDPC codes based on a single-user scenario show the best performance for the assumed complexity restrictions. When a performance-complexity tradeoff is required, LDPC codes based on a single-user scenario as well as punctured LDPC codes seem to be promising candidates, whereas Bi-layer LDPC codes seem to require exhaustive design complexity.

References

[1] E. C. van der Meulen, “Three-terminal communication channels,” Advances in Applied Probability, 1971, vol. 3, p. 121.

[2] T. M. Cover and A. A. El Gamal, “Capacity Theorems for the Relay Channel,” *IEEE Trans. on Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.

[3] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, “On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit,” *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[4] A. Chakrabarti, A. de Baynast, A. Sabharwal, and B. Aazhang, “Low-Density Parity-Check Codes for the Relay Channel,” *IEEE Journals on Selected Areas in Communications*, vol. 25, no. 2, pp. 280–291, Feb. 2007.

[5] P. Razaghi and W. Yu, “Bilayer Low-Density Parity-Check Codes for Decode-and-Forward in Relay Channels,” *IEEE Trans. on Inform. Theory*, vol. 53, no. 10, pp. 3723–3739, Oct. 2007.

[6] J. Ha, J. Kim, and S. W. McLaughlin, “Rate-compatible Puncturing of Low-Density Parity-Check Codes,” *IEEE Trans. on Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[7] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, “Rate-compatible Punctured Low-Density Parity-Check Codes with Short Block Lengths,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 2, pp. 728–738, Feb. 2006.

[8] A. Chakrubarti, A. Subharwal, and B. Aazhang, “Sensitivity of Achievable Rates for Half-Duplex Relay Channel,” in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, New York, USA, Jun. 2005, pp. 970–974.

[9] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor Graphs and the Sum-Product Algorithm,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

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

[11] J. Hagenauer, “Soft-in/soft-out: the Benefits of using Soft Values in all Stages of Digital Receivers,” in *Proceedings 3rd Int. Workshop on DSP Technique Applied to Space Communications, ESTEC*, Noodrdwijk, The Netherlands, Sept. 1992.

[12] S. Y. Chung, T. J. Richardson, and R. Urbanke, “Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[13] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Feb. 1997.

[14] A. Stefanov and E. Erkip, “Cooperative Coding for Wireless Networks,” *IEEE Trans. on Communications*, vol. 52, no. 9, pp. 1470–1476, Sept. 2004.

[15] R. Urbanke, “Degree distribution optimizer for LDPC code ensembles,” http://ipgdemos.epfl.ch/ldpcopt/.