Bilayer LDPC Codes for the Relay Channel

Peyman Razaghi and Wei Yu
Electrical and Computer Engineering Department
University of Toronto
{peyman, weiyu}@comm.utoronto.ca

***Abstract*— This paper describes a methodology for efficient implementation of binning and block-Markov coding for the relay channel using powerful features of low-density parity-check (LDPC) codes. We devise bilayer LDPC codes to approach the theoretically promised rate of the decode-and-forward relaying strategy by incorporating relay-generated random linear parity-bits in a specially designed bilayer graphical code structure. Bilayer density evolution is devised as a novel extension of the standard density evolution algorithm to analyze the performance of the proposed bilayer LDPC code. Based on this bilayer density evolution technique, an EXIT-chart-based code design method using linear programming is developed. While conventional LDPC codes are sensitively tuned to operate efficiently at a certain channel parameter, the proposed bilayer LDPC code is capable of working at two different channel parameters, the signal-to-noise ratio (SNR) at the relay and the SNR at the destination. In this paper, for specific channel parameters, it is demonstrated that a bilayer LDPC code can approach the theoretical decode-and-forward rate of the relay channel within a 0.19 dB gap to the source-relay channel capacity and a 0.34 dB gap to the relay-destination channel capacity.**

I. Introduction

Low-density parity-check (LDPC) codes have proved to be very powerful in approaching the capacity of conventional single user communication channels. The key idea of LDPC codes is to practically implement the random coding theorem of Shannon by enforcing a set of random parity-check constraints on information bits. While random coding is a fundamental element of the single-user information theory, binning is of fundamental importance in multi-user scenarios. In this paper, we explore the possibility of using LDPC codes to practically implement binning and to approach the theoretical results derived by random binning and random coding arguments for an important example of multi-user channels: the relay channel.

In a relay channel, a single source X attempts to communicate to a single destination Y with the help of a relay. The relay receives Y_1 and sends out X_1 based on Y_1. The relay channel is defined by the joint distribution $p(y, y_1|x, x_1)$.

Although the capacity of the relay channel is still an open problem, several ingenious methods have been designed to take advantage of the information available at the relay. The classic work of Cover and El Gamal [1] describes two basic strategies: first, a decode-and-forward strategy in which the relay completely decodes the transmitted message and partially forwards the decoded message using a binning technique to allow the complete resolution of the message at the decoder, and second, a more complex quantize-and-forward strategy in which the relay does not need to decode the source's message. Both coding schemes rely on a clever block-Markov coding strategy in which each coding block consists of simultaneous decoding (or quantizing) of the current block at the relay and the decoding of the previous block at the destination. Further, Cover and El Gamal [1] proved that the decode-and-forward strategy is capacity achieving for a class of degraded relay channels.

Recent interests in wireless ad-hoc and sensor networks have fueled a new surge of research activities on the relay channel [2], [3], [4], [5].Several practical decode-and-forward coding techniques for the relay channel have been developed in [6], [7], [8], where performances approaching 1-1.5dB of the theoretical limit of the decode-and-forward scheme for the relay channel are reported. The coding techniques of [6] and [7] are based on turbo codes while [8] employs LDPC codes. In all these schemes, the relay decodes and retransmits the entire source's codeword. A key feature of our proposed coding scheme is that the relay completely decodes the transmitted codeword, but only forwards some partial information about the codeword by forming random parity bits. This coding scheme is inspired by Cover and El Gamal's [1] original proof for the capacity of the degraded relay channel where the binning strategy is implemented using random parities. The scheme proposed in this paper is related to [9] in which parity bits are used by the relay to enhance diversity. The focus of the present paper is on code design for approaching the capacity.

The main results of our work are as follows. A new bilayer code structure based on LDPC codes has been developed to implement the decode-and-forward strategy of [1]. The proposed bilayer structure allows a single LDPC code to simultaneously approach the capacities of two channels at two different signal-to-noise ratios (SNR), corresponding to the SNR at the relay and the SNR at the destination. We develop a methodology for the design of bilayer LDPC codes by generalizing density evolution [10] and EXIT chart analysis [11], [12] for standard LDPC codes. It is shown that our design methodology can approach the theoretical decode-and-forward rate of the relay channel within a 0.19 dB gap to the source-relay channel capacity and a 0.34 dB gap to the relay-destination channel capacity.

This paper focuses on the Gaussian relay channels at a relatively low SNR so that binary signaling is optimal. We restrict our attention to the block-Markov decode-and-forward strategy, which is optimal for the degraded relay channel but

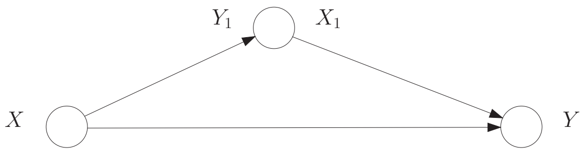

Fig. 1. The relay channel.

is also effective in other cases as well.

II. Block-Markov Decode-and-Forward Coding

A. Coding Scheme for Decode-and-Forward

This section briefly reviews the decode-and-forward strategy of [1, Theorem 1]. In the block-Markov decode-and-forward scheme, transmissions occur in successive blocks and in each block i, the source and the relay send two messages to the destination: the source's data message denoted by $w_i \in \{1, 2, \ldots, 2^{nR}\}$ (which is encoded using the random variable X) and the relay's message $s_i \in \{1, 2, \cdots, 2^{nR_1}\}$ (which is encoded using the random variable X_1.) The source rate, R, is such that the relay is able to decode w_i with an arbitrarily low error probability; however, the destination is unable to uniquely decode w_i because of its poorer channel. The relay's message, s_i, helps the destination decode w_{i-1} in block i by restricting w_{i-1} to be inside a bin of size $2^{n(R-R_1)}$ thus reducing the size of the admissible message space from which the destination should decode the source's message. Let $\mathcal{B} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{2^{nR_1}}\}$ be a random uniform partition of the set $\{1, 2, \cdots, 2^{nR}\}$ into 2^{nR_1} bins of size $2^{n(R-R_1)}$. The relay's message, s_i, is determined as the index of the bin in which w_{i-1} falls, i.e., $w_{i-1} \in \mathcal{S}_{s_i}$.

Random codebooks to transmit s and w are constructed as follows. Assume that in block i, both the source and the relay know s_i; as we shall see, this is a valid assumption, since s_i is determined by w_{i-1}. The source uses different codebooks for different s_i's. To encode w_i, the source utilizes a random codebook $\mathcal{X}(w|s_i)$ of size 2^{nR} generated according to the probability distribution $p(x|x_1)$ and transmits the codeword $\mathbf{x}(w_i|s_i)$ to send w_i. In block i, the relay sends s_i by transmitting the codeword $\mathbf{x}_1(s_i)$ of the random codebook $\mathcal{X}_1(s)$ of size 2^{nR_1} generated according to the probability distribution $p(x_1)$.

In block i, the relay decodes w_i which would be successful as long as:

$$R < I(X; Y_1|X_1). \tag{1}$$

The destination, in block i, first decodes the relay's message s_i which is possible if:

$$R_1 < I(X_1; Y). \tag{2}$$

Upon decoding s_i, w_{i-1} is restricted to the bin $\mathcal{S}_{s_i}$ which is of the size $2^{n(R-R_1)}$. Since w_{i-1} is encoded by a codebook generated according to $p(x|x_1)$, the destination can successfully decode w_{i-1} in block i if R and R_1 satisfy:

$$R - R_1 < I(X; Y|X_1). \tag{3}$$

Inequalities (1), (2), and (3) give the decode-and-forward achievable rate for the relay channel which is also the capacity if the channel is degraded [1, Theorem 1].

B. Gaussian degraded relay channel

Consider the degraded Gaussian relay channel defined by $Y_1 = X + Z_1$ and $Y = Y_1 + X_1 + Z_2$ where $Z_1 \sim \mathcal{N}(0, N_1)$ $Z_2 \sim \mathcal{N}(0, N_2)$ are Gaussian noises.

For this channel, the optimal codebook $\mathcal{X}(w|s_i)$ can be shown to be additive in the sense that codewords of $\mathcal{X}(w|s_i)$, $\mathbf{x}(w_i|s_i)$, can be constructed via $\mathbf{x}(w_i|s_i) = \tilde{\mathbf{x}}(w_i) + \alpha \mathbf{x}_1(s_i)$ where $\mathbf{x}_1(s_i)$ is one of 2^{nR_1} codewords of $\mathcal{X}_1(s)$, $\tilde{\mathbf{x}}(w_i)$ is one of 2^{nR} codewords of the codebook $\tilde{\mathcal{X}}(w)$, and α is a power scaling factor. The optimal value of α is determined in [1, Theorem 5]. See [1, Section IV] for more details.

The binning strategy of the previous subsection can be applied to $\tilde{\mathcal{X}}(w)$ and $\mathcal{X}_1(s)$. A linear codebook can be partitioned into bins by considering the syndromes of codewords with respect to a set of parity equations as bin indices [13]. To implement binning using this idea, the relay in block i decodes the transmitted codeword $\tilde{\mathbf{x}}(w_i)$, generates extra parity bits for $\tilde{\mathbf{x}}(w_i)$ represented by s_{i+1}, and sends it to the destination in the next block using the codebook $\mathcal{X}_1(s)$.

This paper focuses on the design of a new LDPC code structure for $\tilde{\mathcal{X}}(w)$ to implement the described decode-and-forward protocol. First, we note that $\mathcal{X}_1(s)$ can be designed as a conventional LDPC code. However, special considerations are needed for the design of $\tilde{\mathcal{X}}(w)$. Let $\tilde{\mathcal{X}}(w)$ be a linear (n, k_1) LDPC code with a rate of $(n - k_1)/n$. The codebook $\tilde{\mathcal{X}}(w)$ should be a capacity approaching code for the channel between the source and the relay, i.e., (1) should be tight. Let k_2 be the number of randomly-generated extra parity-bits for $\tilde{\mathbf{x}}(w_i)$ generated by the relay and provided to the destination. Then, any $\tilde{\mathbf{x}}(w_i)$ candidate sequence at the destination should satisfy two sets of parities, k_1 zero parities enforced by the source's codebook, and k_2 extra presumably nonzero parity-bits provided by the relay. Thus, $\tilde{\mathcal{X}}(w)$ with the additional k_2 parity checks should form a $(n, k_1 + k_2)$ capacity-approaching code for the source-destination channel.

The objective of this paper is to show that with a modification to the structure of conventional LDPC codes, $\tilde{\mathcal{X}}(w)$ can be designed to approach the achievable rate promised by the decode-and-forward protocol for the relay channel. The main difficulty in the design of $\tilde{\mathcal{X}}(w)$ is that while standard techniques exist for the design of a single LDPC code tuned to work for a specific channel, a new method is required to design a codebook $\tilde{\mathcal{X}}(w)$ that performs well at both the relay SNR, SNR_1, and the destination SNR, SNR_2, with the help of extra relay-provided parities.

III. Bilayer LDPC Codes for the Relay Channel

From the discussion of the previous section, the code structure can be summarized as follows. Let $\tilde{\mathcal{X}}(w)$ have the graphical code structure as shown in Fig. 2, with k_1 zero parity check bits and k_2 extra parity check bits generated by the relay. The source data rate is $(n - k_1)/n$ and the source's codewords

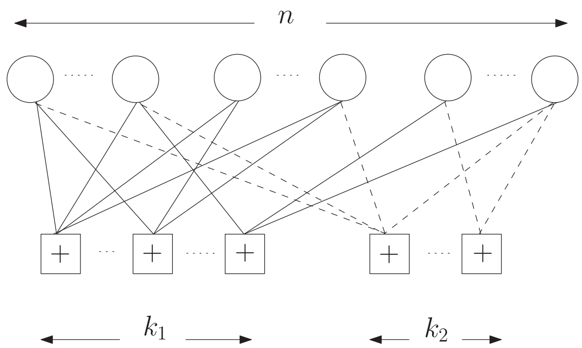

Fig. 2. Bilayer LDPC codes. The solid part corresponds to the subgraph and represents a LDPC code designed for the channel between the source and the relay. The relay decodes the subgraph code and provides extra parity-check bits for the destination (the dashed part). The destination decodes the transmitted codeword over the overall hypergraph.

are enforced to satisfy k_1 zero parity check bits. The relay decodes the source's codeword based on the first k_1 parity bits and generates k_2 extra parity bits which are then transmitted to the destination using a separate codebook, i.e., $\mathcal{X}_1(s)$. The destination first decodes the k_2 extra parity bits sent by the relay and then decodes the source's codeword knowing that it should satisfy k_1 zero parity bits and k_2 extra parity bits generated by the relay. In order to incorporate this protocol, we consider a bilayer structure for the (n, k_1+k_2) LDPC code for $\tilde{\mathcal{X}}(w)$. In this bilayer structure, one layer corresponds to a (n, k_1) capacity approaching LDPC code (for the source-relay channel) consisting of the k_1 zero parity bits and a second layer consists of the k_2 extra parity bits which modifies the first layer in a way that the overall $(n, k_1 + k_2)$ LDPC code is capacity achieving for the source-destination channel.

In the following we present the structure of bilayer LDPC codes and discuss bilayer density evolution and EXIT charts as fundamental design tools.

A. Bilayer LDPC Codes

Let the LDPC code for $\tilde{\mathcal{X}}(w)$ have n variable nodes, k_1 check bits and an additional k_2 check bits (generated by the relay for the destination.) As shown in Fig. 2, the graph corresponding to this code consists of two layers. The *left-graph* or *subgraph* which is directly connected to the left k_1 parities and represents the code designed for the source-relay channel. The *right-graph* is defined to be the part of the graph that is directly connected to the right parities. The right-graph is designed so that it modifies the subgraph in such a way that the resulting *hypergraph* guarantees successful decoding at the destination.

By discriminating *left edges* to be those edges that are connected to the left k_1 parities from *right edges* that are connected to the right k_2 parities, it can be seen that from each variable node, two types of edges may emanate. Therefore, for each variable node of the graph two different degrees are conceivable: the *left degree* which is defined to be the number of left edges connected to the variable and the *right degree* which is defined to be the number of right edges that are connected to the variable.

Let $\lambda_{i,j}$ be the variable degree distribution of hypergraph defined as the percentage of edges in the hypergraph which are connected to a variable node of degree (i, j), i.e., the percentage of edges that have left degree i and right degree j. Note that $i \geq 2$ since no variable of degree less than 2 is allowed in the subgraph and $j \geq 0$ as some of the nodes may only be connected to the left parities. For given $\lambda_{i,j}$'s satisfying $\sum_{2\leq i, 0\leq j} \lambda_{i,j} = 1$ and for a specific set of check degrees, both the subgraph and the hypergraph can be constructed.

In standard LDPC code design, it is common to fix one or at most two different values for check degrees. Some guidelines for choosing appropriate check degrees can be found in [14] and [11]. The rest of this paper focuses on the optimal design of variable degrees $\lambda_{i,j}$ assuming fixed check degrees.

Fixing check degrees, a bilayer code design problem can be formulated as that of finding a doubly indexed distribution $\lambda_{i,j}$ such that the induced subgraph is capacity approaching at SNR_1 and the overall hypergraph is capacity approaching at $\text{SNR}_2 < \text{SNR}_1$.

The degree distribution of the subgraph code can be found as a linear combination of $\lambda_{i,j}$'s as follows:

$$\nu_i = \frac{1}{\eta} \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j} \tag{4}$$

where $0 < \eta < 1$ is the ratio of the total number of edges in the subgraph and the total number of edges in the hypergraph. Assuming a fixed number of check nodes with fixed degrees, the total number of edges in the subgraph and the hypergraph are fixed and therefore η is a constant.

The coefficients ν_i's are related to the code rate between the source and the relay. Let E be the total number of edges in the subgraph. Then, the block length of the code, which is equivalent to the total number of variable nodes in the graph, is given by $E \sum_{i\geq 2} \nu_i/i$; and there are $E \sum_{i\geq 2} \rho_i/i$ left parity check nodes (where ρ_i's denote the fixed left check degree distribution.) Hence, the rate of the source-relay code is given by:

$$R = 1 - \frac{\sum_{i\geq 2} \rho_i/i}{\sum_{i\geq 2} \nu_i/i}. \tag{5}$$

A capacity approaching code for the decode and forward strategy should have an appropriate degree distribution $\lambda_{i,j}$ that maximizes the above rate.

B. Bilayer Density Evolution

Density evolution can be used to analyze the performance of standard LDPC codes [10]. With the use of density evolution, the probability density function of messages is tracked as they pass along the edges through successive decoding iterations. For a given degree distribution, a cut-off channel parameter (the largest noise power under which the code can be successfully decoded) can be found by density evolution.

In a bilayer graph, however, it is necessary to distinguish between the output densities of two variables of the same total

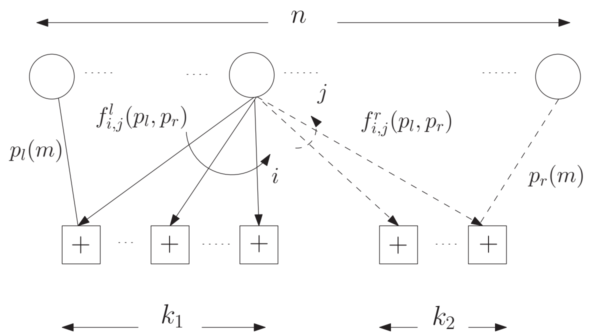

Fig. 3. Bilayer densities and the code structure. A degree (i,j) variable connects to i edges in the subgraph (solid parts) and j edges in the hypergraph (dashed parts). $p_l(m)$ and $p_r(m)$ represent the message densities for the left and right parts of the graph respectively and p_l and p_r denote their corresponding message error probabilities. EXIT charts for degree (i,j) variables are denoted by $f^l_{i,j}(p_l,p_r)$ and $f^r_{i,j}(p_l,p_r)$.

degree but with different left and right degrees. This is because the qualities of messages coming from left check nodes and right check nodes may differ. For example, if the left check degree is 20 and the right check degree is 5, although two variable nodes of degree $(2,9)$ and $(9,2)$ have the same total degree, there is a significant difference between qualities of messages produced by each of them, since the quality of a message produced by a check node of a degree 5 is in general significantly better than the quality of the one that is generated by a degree 20 check node. As a consequence, an extension of the conventional density evolution analysis is required.

An effective way to incorporate the bilayer structure of the underlying graph in the density evolution is to track the evolution of two types of message densities (see Fig. 3): the left density for the messages passing along left edges and the right density for the messages passing along right edges.

Density update at check nodes of a bilayer LDPC code is the same as the standard density updates at check nodes in the conventional density evolution procedure [10] assuming that log-likelihood ratio (LLR) messages are being passed. Let $p_l(m)$ and $p_r(m)$ be the left and right output message densities respectively. Let the channel LLR density be given by $p_c(m)$. At variable nodes, density updates for left edges and right edges are slightly different as compared to variable updates in the standard density evolution. Specifically, for a variable of degree (i,j), $p_l(m)$ and $p_r(m)$ are updated as follows:

$$p_l(m) \leftarrow \left(\otimes^{i-1} p'_l(m)\right) \otimes \left(\otimes^{j} p'_r(m)\right) \otimes p_c(m) \tag{6}$$

$$p_r(m) \leftarrow \left(\otimes^{i} p'_l(m)\right) \otimes \left(\otimes^{j-1} p'_r(m)\right) \otimes p_c(m),\ \ j \geq 1 \tag{7}$$

where $p'_l(m)$ and $p'_r(m)$ denote the input message densities, $\otimes^i$ denotes convolution of order i and by convention $\otimes^0 p(x) = \delta(x)$ where $\delta(x)$ is the Dirac delta function.

C. Extrinsic Information Transfer (EXIT) Charts

We use a graphical tool called extrinsic information transfer (EXIT) chart for LDPC code design. In particular, we use a special type of EXIT charts known as the probability-of-error EXIT chart [11], [12], [15] that describes the relation between the message error probability[1] corresponding to the input message density and the message error probability corresponding to the output message density after one iteration of the density evolution algorithm [10]. EXIT charts are very useful in characterizing the performance of a LDPC code, because they can be used to formulate an approximate successful decoding criterion which is useful in optimizing the degree distribution of a LDPC code.

In this section, we will also use a powerful tool called the elementary EXIT chart [16], [12]. The use of elementary EXIT charts greatly facilitates the LDPC code design process. In this paper, we generalize both EXIT charts and elementary EXIT charts for bilayer LDPC codes.

While EXIT charts characterize the overall performance of a LDPC code, *elementary EXIT charts* give specific information regarding the decoding performance of variable nodes of a certain degree. The elementary EXIT charts are defined as follows. Consider one iteration of the message passing algorithm in which check updates are performed for a given set of input messages at check nodes and subsequently variable updates are applied to the updated messages to obtain a new set of messages at the input of check nodes. (See for example [10] for the definitions of check updates and variable updates in the message passing algorithm.) For a given LDPC code and for a fixed variable degree d, the function relating the input message error probability and the output message error probability along the degree-d edges after one decoding iteration is called the elementary EXIT chart of degree d [16], [12].

There is a linear dependency between EXIT charts and elementary EXIT charts. This is because for irregular codes, the average output message error probability can be obtained via Bayes's rule. Equivalently, the EXIT chart of an irregular code is a linear combination of elementary EXIT charts of various degrees where the coefficients of the linear combination are exactly the variable-degree distribution [15]. This linear combination property makes the error probability EXIT charts a powerful design tool.

Now consider a bilayer LDPC code. The left part of the graph represents a conventional LDPC code and it is straightforward to define the standard EXIT and elementary EXIT charts for it. Let $f^s_i(p)$ be the subgraph elementary EXIT chart of degree i computed corresponding to the source-relay channel parameters, i.e., the left graph when it is decoded in the relay.

Next, consider the entire bilayer graph. Since there are two types of densities involved in the bilayer density evolution, we need to define a multivariable counterpart of elementary EXIT charts for a bilayer graph. Let $f^l_{i,j}(p_l,p_r) : [0,1] \times [0,1] \rightarrow [0,1]$ denote the *left* elementary EXIT chart of degree (i,j) edges where p_l and p_r are the message error probability in the left and right part of the graph. For a given p_l and p_r, $f^l_{i,j}(p_l,p_r)$ represents the message error probability for messages passing along left edges of degree (i,j) after one

[1] A message passing along an edge is said to be correct if it is more biased toward the true value of the corresponding variable node [10].

decoding iteration. Similarly, *right* elementary EXIT chart of degree (i,j), $f^r_{i,j}(p_l,p_r)$, is defined to be the message error probability for messages passing along right edges of degree (i,j).

Since $\lambda_{i,j}$ represents the percentage of degree (i,j) edges in the graph, the overall average output probability of error is given by:

$$p_{out}(p_l,p_r) = \sum_{i\geq 2, j\geq 0} \lambda_{i,j} \frac{f^l_{i,j}(p_l,p_r)i + f^r_{i,j}(p_l,p_r)j}{i+j} \quad (8)$$

which also determines the *average* EXIT chart or more simply EXIT chart of the code. (For $j=0$, $f^r_{i,j}(p_l,p_r)j$ is defined to be zero.) In other words, the EXIT chart (8) is a *linear* combination of elementary left and right EXIT charts with combination factors being $\lambda_{i,j}$'s.

IV. Optimization

EXIT charts can be used to formulate a successful decoding condition for a LDPC code [15]. For simplicity, let's consider first the successful decoding criterion for the subgraph. Consider the output error probability of the subgraph after one decoding iteration. After one iteration, the output error probability can be written as a linear combination of elementary EXIT charts as follows:

$$\sum_{i\geq 2} \nu_i f^s_i(p). \quad (9)$$

In order to decode the codeword successfully, the output error probability should decrease after each iteration. In terms of the above formulation, at any iteration, the output error probability given by (9) should be smaller than the input error probability. This can be formulated as:

$$\sum_{i\geq 2} \nu_i f^s_i(p) < p \qquad \forall\ 0<p<1. \quad (10)$$

The above linear condition gives a simple yet very effective approximate successful decoding condition that can be used to optimize the code rate while ensuring the resulting code can be successfully decoded.

Now, let's consider the successful decoding condition for the hypergraph. Using elementary left and right EXIT charts, a condition similar to (10) can be formulated. Mathematically, using (8) for the average output error probability, the degree distribution coefficients, $\lambda_{i,j}$, should satisfy:

$$\sum_{i\geq 2, j\geq 0} \lambda_{i,j} \frac{f^l_{i,j}(p_l,p_r)i + f^r_{i,j}(p_l,p_r)j}{i+j} \quad < \quad \eta p_l + (1-\eta)p_r$$

for any input left and right error probabilities, $0 \leq p_l, p_r \leq 1$. The above inequality generalizes the open EXIT chart concept to the hypergraph of a bilayer LDPC code. Successful decoding is ensured in (11) by forcing the average error probability of the hypergraph to monotonically decrease as the number of iterations increases.

The overall data rate between the source and the destination in the decode-and-forward coding scheme is determined by the maximum rate at the source which is given by (5). The optimization problem would then be to maximize the source code rate subject to the condition that EXIT charts of the subgraph and the hypergraph, (10) and (11), are both open.

This optimization problem can be formulated as the following linear programming problem:

$$\begin{aligned}
\text{maximize} \quad & 1 - \frac{\sum_{i\geq 2} \rho_i/i}{\sum_{i\geq 2} \nu_i/i} && (11)\\
\text{subject to} \quad & \sum_{i\geq 2} \nu_i f^s_i(p) < \mu_k p, \quad \forall\ 0\leq p\leq 1 && (12)\\
& \sum_{i\geq 2, j\geq 0} \lambda_{i,j} \frac{f^l_{i,j}(p_l,p_r)i + f^r_{i,j}(p_l,p_r)j}{i+j} \\
& \qquad < \mu_k\left(\eta p_l + (1-\eta)p_r\right) \\
& \qquad \forall \quad 0\leq p_l,p_r\leq 1 && (13)\\
& \nu_i = \frac{1}{\eta}\sum_{j\geq 0} \frac{i}{i+j}\lambda_{i,j}. && (14)
\end{aligned}$$

The optimization variables are $\lambda_{i,j}$ and ν_i. All other variables are assumed to be fixed.

In practice, the successful decoding conditions, (12) and (13), do not need to be enforced for all $0\leq p\leq 1$ and all $0\leq p_l,p_r\leq 1$. This is because the message error probabilities p, p_l and p_r correspond to the evolving message densities through successive decoding iterations and are only updated at discrete number of iteration points. As a result, we only need to consider those values of p, p_l, and p_r that correspond to message densities in each decoding iteration.

Strictly speaking, the elementary EXIT charts $f^l_{i,j}(p_l,p_r)$, $f^r_{i,j}(p_l,p_r)$ and $f^s_i(p)$ also depend on $\lambda_{i,j}$, since they are obtained via density evolution which assumes some fixed degree distributions. In practice, the above optimization problem is repeatedly solved, with $f^l_{i,j}(p_l,p_r)$, $f^r_{i,j}(p_l,p_r)$ and $f^s_i(p)$ updated in each step. Because the elementary EXIT charts are slightly modified in each iteration, we find it beneficial to introduce an extra variable μ_k, where $0\leq\mu_k\leq 1$, to compensate the potential inaccuracies in $f^l_{i,j}(p_l,p_r)$, $f^r_{i,j}(p_l,p_r)$ and $f^s_i(p)$. Constraints (12) and (13) with $\mu_k<1$ ensure that the EXIT charts of the subgraph and hypergraph are open by a factor of μ_k at SNR_1 and SNR_2 respectively. An open EXIT chart by a factor of μ_k where k denotes the optimization iteration number, enforces the output probability of error to be less than μ_k times the input error probability. As we solve the sequence of linear programming problems, μ_k is successively increased until it eventually approaches 1.

V. Code Construction

This section presents code construction for a bilayer LDPC code assuming binary-valued codeword sequences for $\tilde{\mathcal{X}}(w)$ and additive white Gaussian noise at both the relay and the destination. The relay's noise power is 0.4356 and the destination's noise power is assumed to be 0.6084. The noise at the destination is assumed to be independent from that of

TABLE I

DESIGNED DEGREE DISTRIBUTION FOR THE BILAYER GRAPH. AN ENTRY (i, j) CORRESPONDS TO $\lambda_{i,j}$, THE PERCENTAGE OF EDGES OF LEFT DEGREE i AND RIGHT DEGREE j.

(i, j)	$j = 0$	$j = 1$	$j = 2$	$j = 3$
$i = 2$	0.1153	0.0623	0	0
$i = 3$	0.1220	0.0921	0	0
$i = 5$	0	0.1897	0	0
$i = 8$	0	0	0.0591	0
$i = 9$	0	0	0.0166	0
$i = 20$	0	0	0.3296	0.0132

the relay. (In this case, the channel is not degraded and the decode-and-forward is suboptimal.)

For these channel parameters, we restrict the code to have a regular left check degree of 18 and a right check degree of 5. These two values for check degrees are determined experimentally by testing several different values for check degrees and studying the behaviors of the different EXIT chart functions. The key to selecting a good check degree is to ensure that there is a wide variety of elementary EXIT charts some of which are widely open and others are overly blocked. This guarantees that a narrowly open overall EXIT chart can be formed by a linear combination of available EXIT charts [14].

In this construction, variable degrees (i, j) are restricted to have i and j less than 20. Note that the performance of LDPC codes improves as the maximum allowed variable degree increases [17]. Using the described iterative linear programming approach of the previous section, an optimized degree distribution sequence $\lambda_{i,j}$ is computed for the given channel parameters. The nonzero degree coefficients are listed in Table I. In this table, entry (i, j) corresponds to the value of $\lambda_{i,j}$ in the resulted bilayer code.

The gap from the theoretical limit can be calculated by comparing the rate of the code with the theoretical limits as expressed in (1) and (3) for binary ± 1 valued codewords. For the source-relay channel, the designed bilayer code is within a 0.19 dB gap to the capacity and has a code rate equal to 0.7520. The corresponding gap for the hypergraph is 0.34 dB and the resulting code rate is 0.6280. The code rate of the subgraph code represents R and the rate of the hypergraph code corresponds to $R - R_1$.

VI. CONCLUSION

Binning is of fundamental importance in multiuser information theory. This paper provides a practical implementation of the binning strategy for the relay channel from a linear coding perspective in which extra parity-check bits are generated at the relay to facilitate the overall communication between the source and the destination.

A key feature of our code design is the construction of a bilayer LDPC code that incorporates the presence of extra parity-check bits from the relay in the decoding process at the destination. Whereas conventional LDPC codes are optimized at a certain SNR, a bilayer LDPC code is tuned to successfully operate at two different SNRs depending on the layer that is being decoded.

In order to analyze the performance of bilayer LDPC codes, the bilayer density evolution is developed as an extension of the conventional density evolution. A new interpretation of elementary EXIT charts based on bilayer density evolution is then applied to the optimization of degree sequence parameters of bilayer LDPC codes via linear programming optimization.

For specific channel parameters, it is demonstrated that a bilayer LDPC code can achieve the theoretical decode-and-forward rate of the relay channel to within a 0.19 dB gap to the source-relay channel capacity and a 0.34 dB gap to the relay-destination channel capacity.

REFERENCES

[1] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.

[2] G. Kramar, M. Gastpar, and P. Gupta, "Capacity theorems for wireless relay channels," in *Allerton Conf. on Commun., Control and Computing*, 2003, pp. 1074–1083.

[3] A. Host-Madsen and J. Zhang, "Capacity bounds and power allocation for wireless relay channel," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2020–2040, June 2005.

[4] R. U. Nabar, H. Bölcskei, and F. W. Kneübuhler, "Fading relay channels: Performance limits and space-time signal design," *IEEE J. Select. Areas Commun.*, vol. 22, pp. 1099–1109, Aug. 2004.

[5] A. Sendonaris, E. Erkip, , and B. Aazhang, "User cooperation diversity – Part I: System description," *IEEE J. Select. Areas Commun.*, vol. 51, pp. 19271938, Nov. 2003.

[6] Z. Zhang, I. Bahceci, and T. M. Duman, "Capacity approaching codes for relay channels," in *Proceedings of IEEE Int. Symp. of Inform. Theory (ISIT)*, 2004, p. 2.

[7] B. Zhao and M. C. Valenti, "Distributed turbo coded diversity for relay channel," *Electronics Letters*, vol. 39, pp. 786–787, May 2003.

[8] M. A. Khojastepour, N. Ahmed, and B. Aazhang, "Code design for the relay channel and factor graph decoding," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, 2004, vol. 2, pp. 2000–2004.

[9] J. P. K. Chu and R. S. Adve, "Cooperative diversity using message passing in wireless sensor networks," in *Proceedings of IEEE Global Telecommun. Conf. (GLOBECOM)*, 2005, vol. 3, pp. 1167–1171.

[10] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 2, pp. 599–618, Feb. 2001.

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[12] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.

[13] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1250–1276, June 2002.

[14] M. Ardakani, *Efficient analysis, design and decoding of low-density parity-check codes*, Ph.D. thesis, University of Toronto, 2004.

[15] M. Ardakani, T. H. Chan, and F. R. Kschischang, "EXIT-Chart properties of the highest-rate LDPC code with desired convergence behavior," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 52–54, Jan. 2005.

[16] M. Ardakani, B. Smith, W. Yu, and F. Kschischang, "Complexity-optimized low-density parity-check codes," in *Allerton conf. on commun., control and computing*, 2005.

[17] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.