This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

On the Designs and Challenges of Practical Binary Dirty Paper Coding

Gyu Bum Kyung and Chih-Chun Wang
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907, USA

***Abstract*—We propose a practical scheme for binary dirty-paper channels. By exploiting the concept of random binning instead of superposition coding, the complexity of the system is greatly reduced. For comparison, the existing approaches require one of the native codes to be of non-uniform *a priori* distribution, which is generally achieved by combining a *symbol mapper* and high-order-alphabet low-density parity-check (LDPC) codes. Using high-order alphabets increases significantly the complexity and the resulting method is not flexible for designing systems of practical channel parameters. In contrast, we propose to implement the random binning concept using only *binary* LDPC and binary convolutional codes. In this work, some design challenges of this random binning approach are identified and addressed. Our systems are optimized by the joint use of density evolution (DE) and the extrinsic information transfer (EXIT) analysis. Simulation results using practical Quasi-Cyclic LDPC codes show that our system achieves similar performance to the state-of-the-art, high-order-alphabet LDPC-based systems while demonstrating significant advantages in terms of complexity and flexibility of system design.**

I. Introduction

Recently, a number of approaches that focus on eliminating inter-user interference (IUI) have been studied in the multiple-input multiple-output (MIMO) antenna systems. Technologies are divided into two categories [1], [2]: linear versus non-linear processing. Channel inversion at the sender and linear minimum mean squared error detection at the receiver are two representative schemes of linear processing. Linear processing incurs minimum computational cost and generally shows good performance at high signal to noise ratio (SNR). However, the performance is away from the capacity for practical SNRs. A representative method of nonlinear processing is dirty paper coding (DPC) [3] based on the general formula with noncausal side information of Gel'fand and Pinsker [4]. Costa showed a surprising fact that the capacity of a Gaussian channel where the transmitter knows the interference noncausally is the same as the capacity of the corresponding interference-free channel. Namely, one can cancel the interference in an information theoretic way without any power penalty.

Enabled by the intriguing results of DPC for Gaussian channels, DPC has since served as footstone of numerous theoretical studies of the Gaussian MIMO broadcast channel [5], [6], and has been the main concepts of most proposals for practical multi-user MIMO broadcast channels. Zamir, Shamai, and Erez suggested a practical solution for Gaussian channels employing the structured binning schemes [7]. A combination of a finer and a coarser *lattice-code* is used, the operation of which resembles shaping and precoding for inter-symbol interference channels [8], [9]. These lattice-code schemes can achieve the dirty paper capacity asymptotically using high-dimensional lattice codes which require high complexity to implement.

In addition to the Gaussian MIMO broadcast channel, DPC has a wide variety of applications in information hiding, data embedding, and watermarking, which focus on the setting of the *binary symmetric channel* instead. In information hiding, covert information can be embedded in the host signal by DPC [10]. The covert information and the host signal have the same roles as those of the transmitted signals and the interference respectively. Once the transmitted signal is carefully designed following the distortion constraint, the covert information can be easily extracted from the host signal at the receiver. There are some theoretical results regarding binary dirty-paper channels. For example, the optimal achievable rate for binary DPC is proved in [11], [12]. However, these theoretical results do not directly translate to practical systems achieving the promised capacity. Bennatan *et al.* proposed a practical superposition-coding-based method for the binary dirty-paper problems in [13].

The main idea of practical dirty paper coding is as follows. Let $\boldsymbol{s}$ denote the interference known to the transmitter but unknown to the receiver. The user would like to communicate an information-bearing codeword $\boldsymbol{c}_1$ to the receiver. A quantization codeword $\boldsymbol{c}_0$ is first chosen by a trellis/convolutional decoder such that $\boldsymbol{c}_0 + \boldsymbol{c}_1$ is the closest to $\boldsymbol{s}$. The difference $\boldsymbol{c}_0+\boldsymbol{c}_1-\boldsymbol{s}$ is then sent by the encoder such that the received signal (plus interference) becomes $(\boldsymbol{c}_0+\boldsymbol{c}_1-\boldsymbol{s})+\boldsymbol{s}+\boldsymbol{n} = \boldsymbol{c}_0+\boldsymbol{c}_1+\boldsymbol{n}$ where $\boldsymbol{n}$ is the additive noise. If an optimal maximum *a posteriori* probability (MAP) decoder is used, the information of $\boldsymbol{c}_1$ can be extracted at the receiver. However, for a practical implementation, if the bit distributions of both codes $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ are uniform on $\{0,1\}$, then no information can be extracted from $\boldsymbol{c}_0 + \boldsymbol{c}_1$ even in a noise-free channel. The reason is that from a bit perspective, the uniformly distributed $\boldsymbol{c}_0$ code is equivalent to a random noise with cross-over probability 0.5. Therefore, no bit-based message-passing decoder can extract any information for $\boldsymbol{c}_1$. Similarly, $\boldsymbol{c}_1$ is equivalent to a $p = 0.5$ noise for $\boldsymbol{c}_0$. Therefore, the BCJR decoder [14] for $\boldsymbol{c}_0$ also has trouble during initialization. Neither of the individual decoders for $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ can be initialized, let alone the joint iterative decoding loop.

One practical solution is to enforce that either $\boldsymbol{c}_0$ or $\boldsymbol{c}_1$

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

should have non-uniform distributed *a priori* distribution to initialize iterative decoding. The existing work for binary dirty-paper channels [13], based on superposition coding, used non-binary low-density parity-check (LDPC) codes and symbol mappers in order to make non-uniform distributed code $\boldsymbol{c}_1$. However, if we use non-binary LDPC codes, the complexity of iterative decoding is prohibitively expensive from the implementation perspective. For example, a **GF**(q) LDPC code can only achieve $q-1$ different non-uniform probabilities $1/q, 2/q, \cdots, (q-1)/q$. Nonetheless, for a general DPC system, a fine granularity of the non-uniform probability is required and one thus has to use a large alphabet-size (say $q \geq 10$). Using a large q increases the system complexity because the complexity of message-passing decoder in **GF**(q) increases by a factor of $q\log(q)$.

In this paper, we propose a practical scheme for binary dirty-paper channels based on random binning. Our scheme employs only binary LDPC and binary convolution codes and their associated, off-the-shelf belief propagation (BP) and BCJR decoders. Without the use of non-uniform codes, we address the problem of decoding $\boldsymbol{c}_0 + \boldsymbol{c}_1$ by combining *edge erasing* with binary LDPC/convolutional codes. This new architecture also allows us to design flexibly the system parameters such as information and quantization code rates to meet the power and data rate requirement of DPC. We optimize our system using density evolution (DE) and the extrinsic information transfer (EXIT) chart. The simulation results using practical Quasi-Cyclic LDPC (QC-LDPC) codes show that our system has a similar performance with the existing superposition-coding-based solution while admitting lower complexity and better flexibility of system design.

It is worth pointing out that another commonly used method to achieve non-uniform codes is *code shaping* [15]. One can modify the results in [13] by using shaping to obtain the non-uniform code $\boldsymbol{c}_1$. If $\boldsymbol{c}_{10}$ and $\boldsymbol{c}_{11}$ are the quantization codeword and the information bearing codeword, the shaped, non-uniform code can be constructed as $\boldsymbol{c}_1 = \boldsymbol{c}_{10} + \boldsymbol{c}_{11}$ This non-uniform code can then be combined with the superposition-coding-based DPC scheme of [13]. The receiver will thus receive $\boldsymbol{c}_0 + \boldsymbol{c}_1 + \boldsymbol{n} = \boldsymbol{c}_0 + (\boldsymbol{c}_{10} + \boldsymbol{c}_{11}) + \boldsymbol{n}$ instead. In this approach, two convolution codes $\boldsymbol{c}_0$ and $\boldsymbol{c}_{10}$ are used in conjunction with the information bearing code $\boldsymbol{c}_{11}$. Since the shaped code $\boldsymbol{c}_1 = \boldsymbol{c}_{10} + \boldsymbol{c}_{11}$ is non-uniform, the BCJR decoder of $\boldsymbol{c}_0$ can be initialized. However, to decode $\boldsymbol{c}_1 = \boldsymbol{c}_{10} + \boldsymbol{c}_{11}$, we still face a similar initialization problem as both $\boldsymbol{c}_{10}$ and $\boldsymbol{c}_{11}$ have uniform distributions and neither decoders of $\boldsymbol{c}_{10}$ and $\boldsymbol{c}_{11}$ can be initialized.[1] The proposed edge-erasing method solves this problem with the use of only one convolution decoder.

II. General Framework of Binary DPC

In this section, we present the general framework of binary DPC with i.i.d. binary additive noise $\boldsymbol{n}$ with cross-over probability p and the interference $\boldsymbol{s}$ known to the sender but unknown to the receiver. The received signal is described by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{s} + \boldsymbol{n}$ while DPC encoder sends the transmitted signal $\boldsymbol{x}$, a function $f(d, \boldsymbol{s})$ of the interference $\boldsymbol{s}$ and information symbol d. The goal of binary DPC is to optimize the transmission rate R subject to a normalized Hamming weight constraint W on $\boldsymbol{x}$.

In [11], [12], the capacity R^* for binary DPC is proved as follows using Gel'fand and Pinsker's results on side-information channels:

$$R^* = \begin{cases} h(W) - h(p) & \text{if } W_0 \leq W \leq 1/2 \\ \alpha W & \text{if } 0 \leq W \leq W_0 \end{cases} \quad (1)$$

where $W_0 = 1 - 2^{-h(p)}$ and $\alpha = \log((1-W_0)/W_0)$.

The capacity R^* can be achieved by the following coset-based *binning* scheme. Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the randomly chosen quantization code and the information bearing code, respectively. In addition, let R_0 and R_1 be the code rates for $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively. The input data $d \in \{1, 2, \cdots, 2^{NR_1}\}$ is transmitted over the channel where N is the codeword length. First, d is mapped to $\boldsymbol{c}_1 \in \mathcal{C}_1$. For each $\boldsymbol{c}_1$, $\boldsymbol{c}_1 + \mathcal{C}_0$ forms a coset (a bin) where $\boldsymbol{c}_1$ is the coset representative. Given the interference $\boldsymbol{s}$, the encoder finds $\boldsymbol{c}_0 \in \mathcal{C}_0$ such that $\boldsymbol{c}_1 + \boldsymbol{c}_0$ in the $\boldsymbol{c}_1 + \mathcal{C}_0$ bin is the closest to $\boldsymbol{s}$ with minimal Hamming distance. Then, the transmitted signal $\boldsymbol{x}$ is simply $\boldsymbol{c}_0 + \boldsymbol{c}_1 - \boldsymbol{s}$, the difference between $\boldsymbol{c}_0 + \boldsymbol{c}_1$ and $\boldsymbol{s}$. With W being the average Hamming weight constraint, we have $E[w_H(\boldsymbol{x})] \leq NW$ for $0 \leq W \leq 0.5$ where $w_H(\cdot)$ denotes the Hamming weight function. Since the quantization code finds the $\boldsymbol{c}_0 + \boldsymbol{c}_1$ that is the closest to $\boldsymbol{s}$ and transmit the difference, to meet the Hamming weight requirement, the quantization code rate R_0 has to satisfy

$$R_0 > 1 - h(W) \quad (2)$$

where $h(\cdot)$ is the binary entropy function [13], [16]. For decoding, the decoder estimates $\hat{d}$ using the received sequence $\boldsymbol{y}$. Therefore, the decoder finds a pair of codewords $\hat{\boldsymbol{c}}_0$ and $\hat{\boldsymbol{c}}_1$ such that $\hat{\boldsymbol{c}}_1 + \hat{\boldsymbol{c}}_0$ is the closest to the received sequence $\boldsymbol{y}$. The $\hat{\boldsymbol{c}}_1$ is then used to find the transmitted symbol $\hat{d}$. The R_1 in the above scheme can achieve the capacity $R^* = h(W) - h(p)$ if $W_0 \leq W \leq 1/2$. For $0 \leq W \leq W_0$, time-sharing is used to achieve the capacity.

III. Binary DPC—the Proposed Practical System

A. The System Model

Similar to that of the traditional forward error control codes, the main difficulty of realizing the performance gain of DPC lies in the construction of efficient encoder and decoder pairs, which is an issue independent to the simple concept discussed previously. We propose a practical binary DPC scheme materializing the coset-based binning discussed previously.

The system model is presented as follows. Fig. 1 shows the block diagram of the proposed scheme using irregular LDPC codes and trellis shaping. Based on the to-be-transmitted data d, the LDPC encoder chooses a codeword $\boldsymbol{c}_1$. We use a random

[1] This initialization problem is unique for binary channels. In [13], the approach of high-order **GF**(q) codes is used only for binary DPC while a more traditional method of binary LDPC plus quantization codes is used for Gaussian channels.

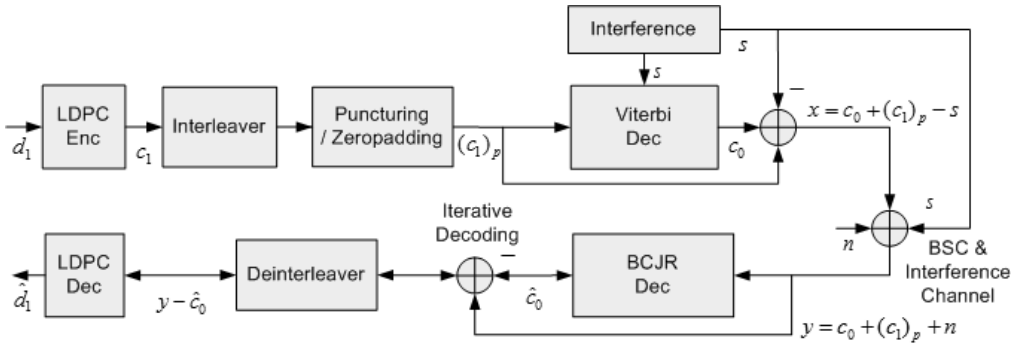

Fig. 1. The system model of the binary dirty paper channel.

interleaver between the LDPC encoder and the Viterbi decoder to reduce the dependence of two encoders. Then, the codeword $\boldsymbol{c}_1 \in \mathcal{C}_1$ is punctured randomly by the fixed portion e, which will be discussed later, and is padded with zeros in those punctured positions. This puncturing is the key element in the proposed system. Let $(\boldsymbol{c}_1)_p$ be the punctured and zero-padded sequence. $(\boldsymbol{c}_1)_p$ and the known interference $\boldsymbol{s}$ are fed into the Viterbi decoder of convolutional codes, which chooses the codeword $\boldsymbol{c}_0$ such that $\boldsymbol{c}_0 + (\boldsymbol{c}_1)_p - \boldsymbol{s}$ is closest to zero. This procedure of the Viterbi decoder is similar to trellis shaping proposed in [15] and we herein minimize the Hamming weight. The transmitter sends the smallest Hamming distance sequence $\boldsymbol{x} = \boldsymbol{c}_0 + (\boldsymbol{c}_1)_p - \boldsymbol{s}$ over the channel.

The interference $\boldsymbol{s}$ and the binary symmetric noise $\boldsymbol{n}$ is added to the transmitted sequence $\boldsymbol{x}$ in BSC and interference channel. Therefore, the received sequence is

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{s} + \boldsymbol{n} = \boldsymbol{c}_0 + (\boldsymbol{c}_1)_p + \boldsymbol{n}. \tag{3}$$

At the receiver, iterative decoding is performed jointly between the BCJR decoder and the LDPC decoder. With puncturing at the transmitter, we now can start BCJR decoding using the received values corresponding to the bits in the punctured part in the BCJR decoder. More explicitly, since the received bits in the punctured positions are $\boldsymbol{y} = \boldsymbol{c}_0 + \boldsymbol{n}$, the log-likelihood ratio (LLR) values at those positions can be set to $(-1)^{y_i} \log \frac{1-p}{p}$ where y_i is the i-th element of $\boldsymbol{y}$ to initialize the whole decoding process. After BCJR decoding, the extrinsic information of the bits in the non-punctured part is then delivered to the LDPC decoder. LDPC decoding is performed with the received bits and the extrinsic LLR information from the BCJR decoder. Then, the LDPC decoder sends the extrinsic information back to the BCJR decoder. Iterative decoding between two decoders continues. Finally, the LDPC decoder outputs $\hat{\boldsymbol{d}}$ as a decoded data after a certain number of iterations.

B. Encoding and Decoding in the Factor Graph

In this subsection, we discuss encoding and decoding based on the factor graph [17] in Fig. 2.

1) Encoding: Fig. 2 illustrates the factor graph of our system assuming an irregular repeat accumulate (IRA) and a convolutional code are used. Our structure can be applied to any type of LDPC codes. We use the blank circles to denote the information variable nodes and the black circles to denote the parity variable nodes. In addition, the squares represent the check nodes. The interleaver Π_1 connects the information variable nodes and the check nodes. The parity variable nodes are then encoded in a dual-diagonal structure using the accumulator. The values of the parity variable nodes correspond to $\boldsymbol{c}_1$ and thus the underlying LDPC codes are non-systematic codes. In addition, the parity bits are permuted according to the interleaver[2] Π_2. Then, the permuted parity bits are punctured and zero-padded in the predetermined positions. We denote the permuted and punctured output as $(\boldsymbol{c}_1)_p$. The Viterbi decoder then selects the codeword $\boldsymbol{c}_0 \in \mathcal{C}_0$ (the circles with cross patterns) such that $\boldsymbol{c}_0 + (\boldsymbol{c}_1)_p$ is the closest to $\boldsymbol{s}$. Then jointly the LDPC and the Viterbi decoder output $\boldsymbol{c}_0 + (\boldsymbol{c}_1)_p$ (the circles with slashes), which will then be directly used to construct the transmitted signal $\boldsymbol{x} = \boldsymbol{c}_0 + (\boldsymbol{c}_1)_p - \boldsymbol{s}$. Since some positions of $(\boldsymbol{c}_1)_p$ are zero-padded, it is equivalent to that some check nodes between the LDPC and the convolutional codes are disconnected from the interleaver Π_2 as shown in Fig. 2. We call this operation "*edge erasing*".

2) Decoding: We use the same factor graph in Fig. 2 for illustration. In the view of the decoder, the circles with slashes represent the received codeword $\boldsymbol{y}$. The received values of the i-th component y_i results in the LLR message $L_{y_i} = (-1)^{y_i} \log \frac{1-p}{p}$. Let I be the set of indices is such that the position i is the index of the section disconnected with the interleaver Π_2. Namely, $y_i = c_{0i} + n_i$ or $y_i = c_{0i} + c_{1i} + n_i$ depending on whether $i \in I$ or not, where c_{0i}, c_{1i}, and n_i are the i-th components of $\boldsymbol{c}_0$, $\boldsymbol{c}_1$, and $\boldsymbol{n}$, respectively. In addition, let us define some LLR edge messages as follows. Let iL be the inbound LLR message to the LDPC parity nodes and let oL be the outbound message from the LDPC parity nodes. Similarly, let iB (oB resp.) be the inbound (outbound resp.) LLR message to (from resp.) the BCJR decoder.

The iterative decoding between the BCJR and the LDPC decoders is described as follows. We use "the LDPC decoder" to refer to the BP decoder. In the case of $i \notin I$, the value of y_i has no influence on the first round of BCJR decoding because both oL and oB are zero at the initial stage. Since if any LLR

[2]The additional interleaver Π_2 is necessary as it facilitates iterative decoding as the interleaver used in turbo codes.

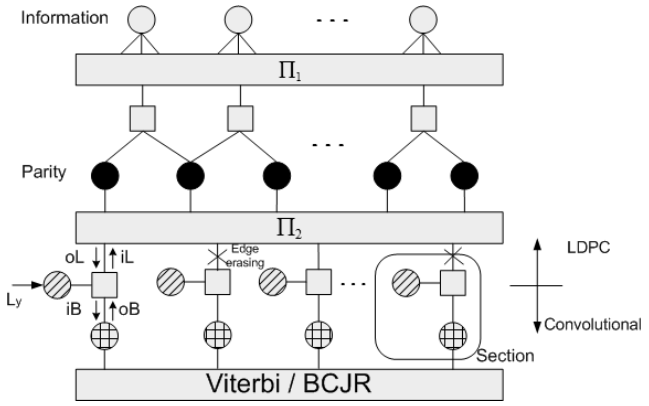

Fig. 2. The factor graph of our system. iL is the inbound LLR message to the LDPC parity nodes and oL is the outbound message from the LDPC parity nodes. Similarly, iB (oB resp.) is the inbound (outbound resp.) LLR message to (from resp.) the BCJR decoder.

input of a check node is zero, the LLR output of the check node is always zero, both iL and iB are zero for those $i \notin I$. Nonetheless, for those $i \in I$, we have $iB = L_{y_i}$ due to edge erasing. These iB messages corresponding to $\{y_i : i \in I\}$ can thus initialize the BCJR decoder.

After the first iteration of the BCJR decoder, the extrinsic information oB is delivered to the LDPC decoder. For those positions $i \notin I$, the messages iL are generated from L_{y_i} and the extrinsic information oB using the standard check node LLR message map. These iLs play a role as *a priori* information to the LDPC decoder. For those $i \in I$, we simply set the LLR message $iL = 0$. After a certain number[3] of iterations of LDPC decoding based on iL, the LDPC decoder generates the extrinsic information oL, which is transmitted back to the BCJR decoder. The input messages iB of the BCJR decoder can then be computed from L_{y_i} and the extrinsic information oL. These iBs perform a role as *a priori* information to the BCJR decoder. The BCJR decoder can then start a second-round decoding based on iBs for those $i \notin I$ and the L_{y_i} for those $i \in I$.

IV. Code Design

A. Choose the system parameters

The constraint of the system is the Hamming weight constraint W. For any BSC with cross-over probability p and weight constraint W, we choose the smallest R_0 such that the quantization code R_0 results in Hamming distortion less than W (cf. the optimal R_0 is governed by (2)). Rate R_1 is chosen by (1). *Note by varying the rate R_0 (such as using punctured codes), our system is flexible and can easily handle different weight constraints W.*

For comparison, for a given DPC system using non-uniform code $\boldsymbol{c}_1$. The normalized Hamming weight σ of $\boldsymbol{c}_1$ must satisfy $W = \sigma * p$ in order to achieve the capacity where $a * b \triangleq a(1-b) + (1-a)b$. In the existing work [13], a high-order $\mathbf{GF}(q)$ LDPC code is used with a symbol mapper to generate a non-uniform code $\boldsymbol{c}_1$, which has σ being $1/q$ to $(q-1)/q$. Therefore, if $\mathbf{GF}(4)$ is used for a BSC with cross-over probability $p = 0.1$, the system can support only $W = 1/4 * p = 0.3$.[4] If a $\mathbf{GF}(8)$ code is used with $p = 0.1$, then only three weights are supported: $W = 0.2$ ($\sigma = 1/8$), $W = 0.3$ ($\sigma = 2/8$), and $W = 0.4$ ($\sigma = 3/8$). To flexibly support different W values, very high-order $\mathbf{GF}(q)$ has to be used, which significantly increases the complexity of the system. Note that in our proposed scheme, only binary LDPC codes are used, which admits great complexity advantages over any non-binary LDPC codes.

B. Code Optimization

We perform code optimization based on the joint use of DE [18] and the EXIT chart [19]. We use DE to analyze the performance on the LDPC-code side of our system, which tracks the LLR message distributions in the LDPC decoder. The EXIT chart is then used to analyze the BCJR decoder and the joint iteration between the BCJR and LDPC decoders. We define e as the ratio between the cardinality of the set of this partial received messages and the number of parity nodes, that is, $e = |I|/N$. It is worth pointing out that the erasure percentage e is critical to the system performance. The larger value e is better initialization of the BCJR decoder as more bits are assigned with nonzero LLR ($iB \neq 0$). However, the large e value also prevents the LDPC decoder from receiving decoded soft bit values from the BCJR decoder. A balance needs to be structured between these two effects.

We first describe how to combine DE with the EXIT chart analysis. We use oL and iL as the input and the output of the EXIT chart that will describe the convergence of the system. Assuming oL is always Gaussian distributed with mean and variance $(\mu, 2\mu)$ for some μ. (As verified in our decoding simulation, even with a BSC, the distribution of oL can be approximated as Gaussian.) Then we start by plotting the EXIT curve of the BCJR decoder according to [19]. More explicitly, the mutual information of the LLR messages is calculated as follows:

$$\begin{aligned} I(X;Y) &= h(X) - h(X|Y) = 1 - h(X|Y) \\ &= 1 - \int_{-\infty}^{+\infty} \log_2 \frac{e^m + 1}{e^m} P(m|X=0)dm \end{aligned} \tag{4}$$

where m denotes the LLR messages, that is, $m = \frac{\log_2(Y|X=0)}{\log_2(Y|X=1)}$. We can then record the mutual information at the output LLR message iL of the BCJR decoder. The EXIT curve of the BCJR decoder is obtained by connecting different (oL, iL) mutual information pairs.

For the EXIT curve of the LDPC code, we use the mutual information at iL to generate a Gaussian message distribution, which is used to initialize the DE. Depending on the predefined number of iterations performed within the LDPC-side of the decoder, we perform the same number of DE iterations

[3]This number can be chosen adaptively as explained in Section IV-B.

[4]The cases $\sigma = 2/4$ and $3/4$ are less interesting as they can be achieved by ordinary binary codes or by switching the roles of 0s and 1s.

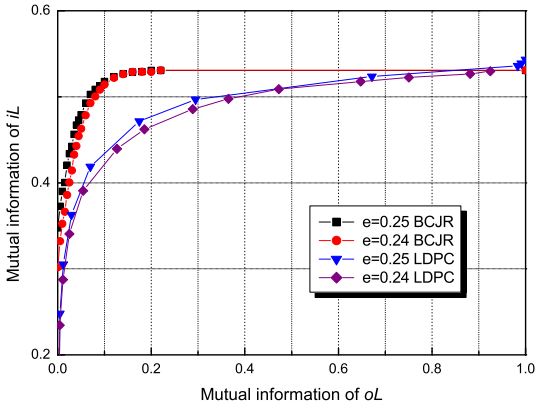

Fig. 3. The EXIT curves of the BCJR decoder and the LDPC decoder for different percentages of edge-erasing.

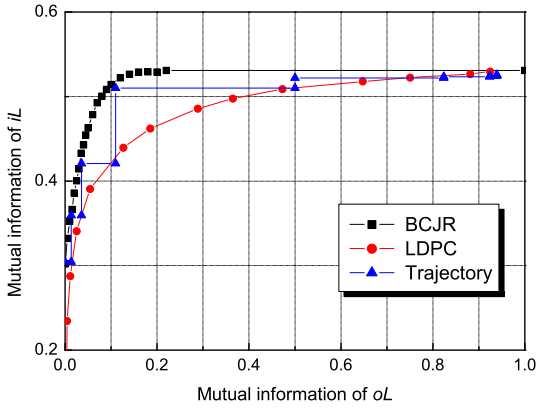

Fig. 4. The EXIT chart trajectory of the system.

within the LDPC code. The density of the final LLR output is then the distribution of the LLR messages oL. By computing the corresponding mutual information of oL by (4), we can connect the corresponding (iL, oL) mutual information pairs to obtain the EXIT curve for LDPC codes.

The EXIT chart can be used to estimate the threshold p^* and to optimize the e value. Since the EXIT curves of the BCJR and LDPC decoders move when different cross-over probability p is considered, by selecting the largest p values such that the two curves do not cross each other, we can have an estimate of the decoding threshold p^*. On the other hand, given p, for each e value, we can obtain two EXIT curves (one for the BCJR decoder and one for the LDPC decoder). Fig. 3 illustrates two such pairs of curves for $e = 0.24$ and $e = 0.25$ accordingly. We choose an e value such that the two curves are the farthest apart.

Since we have to search for a much larger design space when optimizing the degree distributions, we modify the EXIT chart analysis in the following way. The EXIT curve of the BCJR decoder is computed in the same way as described previously. Nonetheless, instead of simply recording the output mutual information by (4), we also record the distribution of the output iL resulted from the BCJR decoder.[5] We then use the computed pmf as an input of DE. Then, we perform the pre-defined number of DE iterations and obtain the final distribution of the LLR output oL. By computing the corresponding mutual information of oL by (4), we can use the previously computed EXIT curve of BCJR to quickly find the pmf of iL in the next round. This new pmf of iL can then be used for another round of DE. In this manner, we can test efficiently that given a degree distribution, whether the above EXIT curve and DE computation converge to successful decoding within the given number of total iterations. We then use the differential evolution method [20] to optimize the degree distribution.

To verify that the performance predicted by the above optimization methodology indeed matches the performance of real decoder. We record the LLR distribution under real BCJR+LDPC decoding with Monte-Carlo simulation and then convert to the correspond mutual information by (4). Fig. 4 shows such a trajectory of the real system. We obtain the pmfs of oL and iL with $p = 0.1$ and $e = 0.24$ through Monte-Carlo simulation and calculated the mutual information of the pmfs by (4). We also compare the trajectory with the BCJR EXIT curve and the LDPC EXIT curve in Fig. 3 which are obtained from Gaussian approximation of the inputs as described previously. The zigzag trajectory fits the two EXIT curves except for a small number of points.

This trajectory curve can also be used to decide how many iterations one should proceed within the LDPC decoder. Usually, the small number of iterations (e.g. 10-15) is required at the initial stage of iterative decoding. However, the required iterations will be increased as the iterative decoding between the BCJR decoder and the LDPC decoder proceeds. By choosing different numbers of LDPC iterations in the initial and final stages of decoding, we can control the total number of LDPC iterations (summing over both the initial and final stages) to be roughly 100-150, which is comparable to that of practical LDPC decoders for i.i.d. channels.

V. Simulation and Discussions

In our simulation, we use irregular LDPC codes as channel codes $\mathcal{C}_1$ and convolutional codes as quantization codes $\mathcal{C}_0$. Specifically, we use QC-LDPC codes [21] and the associated efficient encoding and decoding commonly used in the practical system such as 802.16e [22].

For the sake of comparison, we use the same parameters that are used in [13] such as the code rates (R_0, R_1) = (0.125, 0.36), the codeword length ($N = 10^5$), and the generator polynomials of a convolutional code given by

$$(2565, 2747, 3311, 3723, 2373, 2675, 3271, 2473). \quad (5)$$

[5] As only quantized messages will be used in practice, the distribution of iL can be described by a pmf.

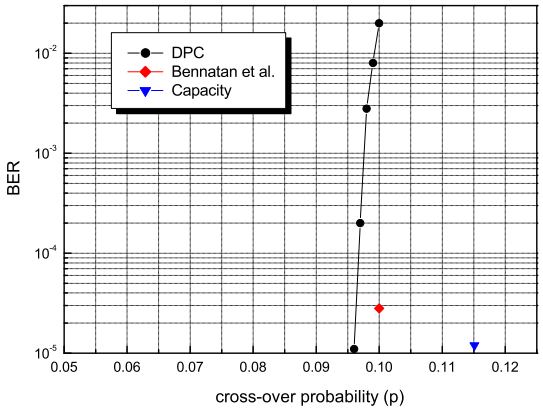

Fig. 5. The bit error rate of the system.

In addition, our optimized binary irregular LDPC code has the degree distributions as follows:

$$\lambda(x) = 0.53x + 0.21x^2 + 0.01x^3 + 0.25x^9, \\ \rho(x) = 0.2x^2 + 0.8x^3. \tag{6}$$

From (2), we obtain the optimal power constraint $W^* = 0.295$. In our Monte-Carlo simulation, the average value of W resulted from our convolutional quantization code is 0.308. The bit-error rates (BER) curve of our binary DPC is compared with the results of [13] in Fig. 5. The BER of our code is comparable to their results. At BER$= 3 \times 10^{-5}$, the threshold of our system is $p^* = 0.0965$ while the superposition-coding + **GF**(4)-based system [13] achieves $p^* = 0.1$. It is worth noting that our system is based on practical binary QC-LDPC structured codes rather than the capacity approaching random ensemble. In general, the performance of QC-LDPC codes is a bit worse than the randomly-optimized irregular LDPC codes since QC-LDPC codes must have special structure for efficient encoding and decoding. Moreover, the degree distributions of QC-LDPC codes are coarsely quantized (see (6)) to ensure feasibility. With a bit sacrifice (less than 5%) in the system performance, our practical DPC scheme can be encoded and decoded using the efficient LDPC encoder/decoders [23] and the BCJR/Viterbi decoder. For reference, the capacity of this code is $p^*_{DPC} = 0.1151$ by (1).

VI. Conclusions

We have proposed a practical scheme for binary dirty-paper channels. Our approach is based on random binning instead of superposition coding, the latter of which requires one of the native codes to be of non-uniform *a priori* distribution. The non-uniform code is generally achieved by combining a symbol mapper and non-binary LDPC codes, which induces prohibitively high complexity and is less flexible for practical values of system parameters. We have suggested using binary LDPC codes and edge erasing with the random binning concept which have the advantages for complexity. Moreover, we can choose system parameters flexibly, which is important in the practical system. We have also provided the code design combining both the EXIT chart and the DE analysis. Our simulation results have demonstrated similar performance to that of state-of-the-art superposition-coding-based binary DPC scheme.

References

[1] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, "An introduction to the multi-user MIMO downlink," *IEEE Communications Megazine*, vol. 42, no. 10, pp. 60–67, Oct. 2004.

[2] D. Gesbert, M. Kountouris, R. W. Heath, Jr., C.-B. Chae, and T. Salzer, "Shifting the MIMO paradigm: From single user to multiuser communications," *IEEE Sig. Proc. Mag.*, vol. 24, pp. 36–46, Oct. 2007.

[3] M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[4] S. Gel'fand and M. Pinsker, "Coding for channel with random parameters," *Probl. Contr. Inform. Theory*, vol. 9, no. 1, pp. 19–31, Jan. 1980.

[5] T. Cover, "Broadcast channels," *IEEE Trans. Inform. Theory*, vol. 18, no. 1, pp. 2–14, Jan. 1972.

[6] G. Caire and S. Shamai, "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1691–1706, July 2003.

[7] 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.

[8] U. Erez, S. Shamai, and R. Zamir, "Capacity and lattice strategies for canceling known interference," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3820–3833, Nov. 2005.

[9] U. Erez and S. ten Brink, "A close-to-capacity dirty paper coding scheme," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3417–3432, Oct. 2005.

[10] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[11] R. Barron, B. Chen, and G. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1159–1180, May 2003.

[12] S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding with side information," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.

[13] A. Bennatan, D. Burshtein, G. Caire, and S. Shamai, "Superposition coding for side-information channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 1872–1889, May 2006.

[14] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[15] G. D. F. Jr., "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 281–300, Mar. 1992.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[17] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[18] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[19] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Communications*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[20] K. Price and R. Storn, "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.

[21] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[22] "IEEE P802.16e-2006 draft standards for local and metropolitan area networks part 16: Air interface for fixed broadcast wireless access systems," *IEEE Standard 802.16e*, Feb. 2006.

[23] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.