

# Analysis of Low-Density Parity-Check Codes for the Gilbert–Elliott Channel

Andrew W. Eckford, *Member, IEEE*, Frank R. Kschischang, *Senior Member, IEEE*, and Subbarayan Pasupathy, *Fellow, IEEE*

**Abstract**—Density evolution analysis of low-density parity-check (LDPC) codes in memoryless channels is extended to the Gilbert–Elliott (GE) channel, which is a special case of a large class of channels with hidden Markov memory. In a procedure referred to as estimation decoding, the sum–product algorithm (SPA) is used to perform LDPC decoding jointly with channel-state detection. Density evolution results show (and simulation results confirm) that such decoders provide a significantly enlarged region of successful decoding within the GE parameter space, compared with decoders that do not exploit the channel memory. By considering a variety of ways in which a GE channel may be degraded, it is shown how knowledge of the decoding behavior at a single point of the GE parameter space may be extended to a larger region within the space, thereby mitigating the large complexity needed in using density evolution to explore the parameter space point-by-point. Using the GE channel as a straightforward example, we conclude that analysis of estimation decoding for LDPC codes is feasible in channels with memory, and that such analysis shows large potential gains.

**Index Terms**—Density evolution, estimation-decoding, Gilbert–Elliott channel, low-density parity-check codes.

## I. INTRODUCTION

LOW-density parity-check (LDPC) codes [1], a class of codes with very sparse parity-check matrices, have generated a great deal of interest as a result of their excellent error-correction performance using simple iterative decoding algorithms. These codes may be decoded by applying the sum–product algorithm (SPA) in a factor-graph representation of the code, as described, e.g., in [2]. Because of the presence of cycles in the graph, this algorithm does not provide minimum symbol-error-rate performance; nevertheless, simulations involving transmission over memoryless channels demonstrate that LDPC codes can be highly effective. Indeed, irregular LDPC codes are among the best known block codes, in the sense that they closely approach the Shannon limit [3], [4]. Density evolution (DE) analysis [3], [5] provides performance thresholds for LDPC codes of a given construction, establishing a region of channel parameters over which iterative decoding is successful (symbol-error-rate bounded below a given value) in the limit of long block length.

Manuscript received October 25, 2002; revised May 28, 2005. The material in this paper was presented in part at the 41st Annual Allerton Conference on Communications, Control and Computing, Monticello, IL, October 2003.

The authors are with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: andrew.eckford@utoronto.ca; frank@comm.utoronto.ca; pas@comm.utoronto.ca).

Communicated by A. Kavčić, Associate Editor for Detection and Estimation. Digital Object Identifier 10.1109/TIT.2005.856934

The application of LDPC codes to channels with memory has attracted increasing interest in the literature. The conventional approach is illustrated in [6], where an LDPC code was applied to a correlated Rayleigh-fading channel. The authors noted that, for sufficiently long block length, the random construction of the code acts implicitly as an interleaver, destroying any latent channel memory effects. However, many recent papers have pointed out that joint estimation-decoding algorithms can result in significantly improved performance for channels with memory. Early work in this area includes that of Wiberg [7], who discussed the inclusion of estimation on the same factor graph as the LDPC decoder, and that of Worthen and Stark [8], who implemented such an integrated estimation-decoding system for a block fading channel. The theory of adapting the factor graph representation of a decoder to incorporate the random behavior of a channel with memory was advanced in [9], [10]. More recently, a joint estimation-decoding scheme was proposed for the Gilbert–Elliott (GE) channel for turbo codes in [11] and LDPC codes in [12]. In particular, [12] shows empirically that joint estimation-decoding strategies can result in reliable communication at rates above the capacity of the equivalent memoryless channel (though, of course, below the capacity of the channel with memory). A similar estimation-decoding algorithm for more general Markov noise sources was presented in [13]. DE was used to analyze an estimation-decoding scheme for LDPC codes in an intersymbol interference channel in [14], and to compare the performance of LDPC codes and other turbo-like codes in partial response channels in [15]. All of this work shows that the performance of LDPC codes continues to be excellent, and that taking channel memory into account leads to improvements in performance.

The DE algorithm establishes bounds on the ultimate performance of LDPC codes in the limit of long block length, allowing the calculation of a threshold in a memoryless channel with a one-dimensional parameter space. The DE algorithm implements a decision function, taking channel parameters as inputs, and determining whether or not the decoder achieves a very small probability of error for these parameter values. As such, the DE technique can only be used to probe *individual* points within the space of channel parameters. Efficient characterization of this space requires knowledge of the implications of the status of decoding at each point in as many dimensions of the channel parameter space as possible. If the decoder achieves small probability of error for one particular channel with memory, ideally this point should induce a multi-dimensional *decoding region* within the parameter space within which every set of channel parameters also achieves the same

or smaller probability of error. For instance, if a memoryless channel having a single parameter is *monotonic*, then an interval of parameter values is induced by each point for which small probability of error is achieved [5], resulting in a threshold. Similarly, if the DE algorithm returns the result that some point has probability of error bounded away from zero, this point should induce a region in which every point has the same or greater probability of error. In this way, the parameter space is partitioned into a decoding region, a nondecoding region, and an uncertain region. The size and shape of the uncertain region then guides the selection of new points to test using the DE algorithm.

In this paper, we present a DE-based analysis of LDPC decoding over the GE channel. The GE channel is used for two reasons: first, it is one of the simplest nontrivial channels with memory, and second, it is a coarse approximation for more practical and complicated channels with memory, such as fading channels. Our objectives are as follows:

- to make DE analysis feasible in a channel with memory, by providing an efficient means of characterizing the parameter space; and
- to show that joint estimation decoding in the GE channel is a *quantifiably* good strategy under a wide range of parameters.

The contributions of this work include the following results. We prove in Section II that density evolution analysis is applicable to GE channels by showing that various necessary conditions are satisfied. In Section III, we state several theorems which characterize the space of GE channel parameters, showing that any point for which estimation decoding is known to converge to small probability of error induces a region of successful decoding in the parameter space, generalizing the notion of thresholds from [5]. Conversely, we show that a point for which estimation decoding does not converge to small probability of error induces a region of unsuccessful decoding. In Section IV, we derive the density evolution algorithm for the GE channel. In Section V, we provide results from the DE algorithm demonstrating the partitioning of GE channel parameters into decoding, nondecoding, and uncertain regions. The decoding regions are shown to be significantly larger than the regions of parameters where estimation decoding converges to small probability of error under the assumption of perfect interleaving (i.e., the assumption that all channel memory is destroyed). All these results indicate that analysis of LDPC decoding is feasible in channels with memory, and that large gains are possible using joint estimation-decoding techniques.

## II. SYSTEM MODEL AND PREREQUISITES

In this section, we discuss the necessary channel and decoder models to motivate the remainder of the paper. We begin by introducing the factor-graph model for an estimator decoder for LDPC codes in the GE channel. Subsequently, we show that this decoder satisfies the necessary conditions for DE analysis.

Throughout this paper, random variables will be denoted with upper case letters, and specific values from the corresponding sample space with corresponding lower case letters. We will use the notation  $\Pr(E)$  to represent the probability of an event  $E$ , and  $p_X(x)$  to represent the probability mass function (pmf) of

a (discrete) random variable  $X$ . Thus,  $p_X(x) := \Pr(X = x)$ . Likewise, the probability density function (pdf) of a continuous random variable  $X$  will be denoted as  $f_X(x)$ . When no confusion can arise, we will let the argument of the pmf or pdf identify the corresponding random variable, writing  $p(x)$  for  $p_X(x)$  and  $p(y)$  for  $p_Y(y)$ . Thus,  $p(x)$  and  $p(y)$  are, in general, *different* functions. Likewise, we will write  $p(y|x)$  for

$$p_{Y|X}(y|x) := \Pr(Y = y | X = x).$$

Vectors will be denoted in bold letters, e.g., we may have  $\mathbf{X} = (X_1, \dots, X_n)$  and

$$p_{\mathbf{X}}(\mathbf{x}) = \Pr(X_1 = x_1, \dots, X_n = x_n)$$

where  $\mathbf{x} = (x_1, \dots, x_n)$ .

### A. LDPC Estimation Decoding

The GE channel is a binary-input, binary-output channel in which the channel output vector  $\mathbf{Y} \in \{0, 1\}^n$  in response to the channel input vector  $\mathbf{X} \in \{0, 1\}^n$  is given as

$$\mathbf{Y} = \mathbf{X} \oplus \mathbf{Z}$$

where  $\mathbf{Z} \in \{0, 1\}^n$  is a noise sequence, and  $\oplus$  denotes componentwise modulo-2 addition. The noise sequence  $\mathbf{Z}$  arises from a two-state hidden Markov process with state sequence  $\mathbf{S} \in \mathcal{S} := \{B, G\}^n$ , such that  $\Pr(Z_i = 1 | S_i = s_i) = \eta_{s_i}$ , where we define  $\eta_G \leq \eta_B \leq 1/2$ . The state  $B$  represents a “bad” binary-symmetric channel (BSC) with relatively large crossover probability, whereas state  $G$  represents a “good” BSC with relatively smaller crossover probability. The state transition probabilities are given by

$$b := \Pr(S_{i+1} = B | S_i = G)$$

and

$$g := \Pr(S_{i+1} = G | S_i = B)$$

resulting in marginal steady-state probabilities of  $\Pr(S_i = B) = b/(b + g)$  and  $\Pr(S_i = G) = g/(b + g)$ , and an average inversion probability of

$$\bar{\eta} := \Pr(Z_i = 1) = \frac{b\eta_B + g\eta_G}{b + g}. \quad (1)$$

Throughout this paper, we assume that the GE Markov process is in steady state. We define the GE parameter space  $\mathcal{G}$  as the set of all 4-tuples  $(b, g, \eta_B, \eta_G)$  representing points in the GE parameter space, such that  $0 \leq \eta_G \leq \eta_B \leq 1/2$  (i.e., noninverting channels) and  $0 < \{b, g\} < 1$  (i.e., nondeterministic channels). In this paper, we restrict our analysis to the parameter space represented by  $\mathcal{G}$ .

Now let the channel inputs be codewords from a given binary LDPC code  $\mathcal{C}$  with an  $m \times n$  parity-check matrix  $\mathbf{H}$ . We will assume that the LDPC code is regular with check degree  $d_c$  and variable degree  $d_v$ , so that each row of  $\mathbf{H}$  contains exactly  $d_c$  ones and each column of  $\mathbf{H}$  contains exactly  $d_v$  ones. The  $k$ th row of  $\mathbf{H}$  is denoted  $\mathbf{h}^k$ , and the transpose of  $\mathbf{h}^k$  is denoted  $(\mathbf{h}^k)^T$ .

Let  $h : \{0, 1\}^n \rightarrow \{0, 1\}$  be a codeword indicator function, where  $h(\mathbf{x}) = 1$  if  $\mathbf{x} \in \mathcal{C}$  and  $h(\mathbf{x}) = 0$  otherwise. Let  $h_k : \{0, 1\}^n \rightarrow \{0, 1\}$  be the indicator function for the  $k$ th row of  $\mathbf{H}$ , defined so that  $h_k(\mathbf{x}) = 1$  if and only if  $\mathbf{x}(\mathbf{h}^k)^T = 0$ , i.e., if and only if  $\mathbf{x}$  satisfies the parity-check equation represented by the  $k$ th row of  $\mathbf{H}$ . The function  $h_k$  may be considered as a function only of those codeword positions that actually participate in this parity-check equation, i.e., we may write  $h_k$  as  $h_k(\mathbf{x}_k)$ , where  $\mathbf{x}_k$  represents the restriction of  $\mathbf{x}$  to those positions where ones appear in  $\mathbf{h}^k$ . Since  $\mathbf{x}$  is a valid codeword if and only if it satisfies all parity-check equations, we have

$$h(\mathbf{x}) = \prod_{k=1}^m h_k(\mathbf{x}_k);$$

a factorization that can be represented using a factor graph as explained in [2]. Assuming that each codeword of  $\mathcal{C}$  is equally likely to be selected by the transmitter, we have  $p(\mathbf{x}) = h(\mathbf{x})/|\mathcal{C}|$ , where  $|\mathcal{C}|$  denotes the number of codewords of  $\mathcal{C}$ .

Likewise, the pmf for the state sequence  $\mathbf{S}$  can be factored, since it is a Markov chain, as

$$p(\mathbf{s}) = p(s_1) \prod_{j=1}^{n-1} p(s_{j+1}|s_j).$$

The joint pmf of the channel output  $\mathbf{Y}$ , codeword  $\mathbf{X}$ , and state sequence  $\mathbf{S}$  is therefore given by

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}, \mathbf{s}) &:= p(\mathbf{y} | \mathbf{s}, \mathbf{x}) p(\mathbf{s}) p(\mathbf{x}) \\ &= \frac{1}{|\mathcal{C}|} \left( \prod_{i=1}^n p(y_i | x_i, s_i) \right) \\ &\quad \cdot \left( p(s_1) \prod_{j=1}^{n-1} p(s_{j+1} | s_j) \right) \left( \prod_{k=1}^m h_k(\mathbf{x}_k) \right). \end{aligned} \quad (2)$$

Using this model and message-passing decoding using the SPA, since there are loops in the factor graph, an estimate for the *a posteriori* probability  $p(x_i | \mathbf{y})$  for each of the transmitted symbols  $x_i$  may be obtained through marginalization and subsequent normalization, using the definition of conditional probability. This probabilistic model results in a factor graph formed by connecting the LDPC factor graph and the GE Markov chain factor graph, so that edges are created to connect the appropriate symbol-variable nodes and channel factor nodes. This GE-LDPC decoder is depicted graphically in Fig. 1. For clarity, we will refer to variable and factor nodes in the LDPC subgraph as *symbol-variable nodes* and *parity-check nodes*, respectively, and variable and factor nodes in the Markov subgraph as *state-variable nodes* and *state-factor nodes*, respectively.

### B. Sum-Product Messages

In terms of message passing using the SPA, the two messages in the LDPC subgraph, which are messages from symbol-variable nodes to parity-check nodes and *vice versa*, are now joined by four new messages, depicted in Fig. 2: the vector *forward message*, represented by  $\mathbf{A}$ ; the vector *backward message*, represented by  $\mathbf{B}$ ; the scalar *channel message*, represented by  $C$ ; and the scalar *extrinsic message*, represented by  $D$ .

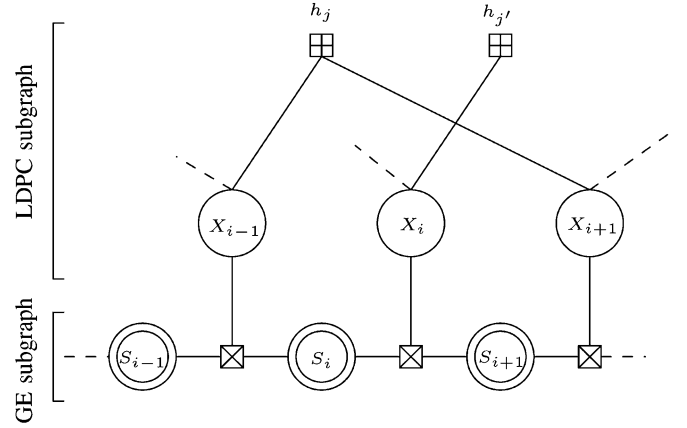


Fig. 1. GE-LDPC decoder graph.

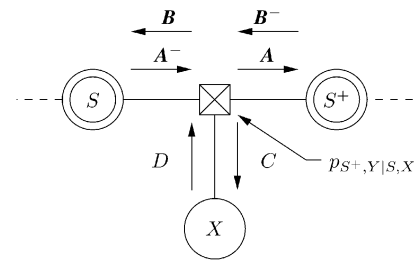


Fig. 2. A depiction of the message flow through the Markov subgraph.

We first define some useful quantities for message calculation. Let  $\mathbf{u}_k$  be the all-one column vector with  $k$  components. Let  $\mathbf{P}$  be the state transition probability matrix, so that

$$\mathbf{P} = \begin{bmatrix} 1-b & b \\ g & 1-g \end{bmatrix}.$$

Let  $\sigma : \{0, 1\} \rightarrow \{1, -1\}$  be the function that translates between the two types of binary random variables, where  $\sigma(Y) = 1$  if  $Y = 0$  and  $\sigma(Y) = -1$  if  $Y = 1$ . Let  $\gamma : \mathbb{R} \times \{0, 1\} \rightarrow [0, 1]$  be defined as

$$\gamma(\lambda, y) = \frac{1}{2} \left[ 1 + \sigma(y) \tanh \left( \frac{\lambda}{2} \right) \right]. \quad (3)$$

If  $\lambda$  is the log-likelihood ratio  $\log(p(0)/p(1))$ , then  $\gamma(\lambda, y)$  returns  $p(y)$ . Furthermore, let  $\mathbf{N} = \text{diag}[\eta_G \eta_B]$ , and let

$$\mathbf{E}(\lambda, y) = \mathbf{N}(1 - \gamma(\lambda, y)) + (\mathbf{I} - \mathbf{N})\gamma(\lambda, y)$$

where  $\mathbf{I}$  is the  $2 \times 2$  identity matrix.

The forward message  $\mathbf{A}$  and the backward message  $\mathbf{B}$  carry summarized information about a channel state  $S$ . Under the SPA, the messages are functions of  $S \in \{B, G\}$ , so  $\mathbf{A}$  and  $\mathbf{B}$  are two-element vectors, with each element corresponding to a point of the function. In both cases, the first element of the vector corresponds to the  $G$  state.

Referring to Fig. 2, the message  $\mathbf{A}$  is calculated from an input forward message  $\mathbf{A}^-$  extrinsic information message  $D$ ,

and channel output  $Y$ . Using the SPA, following some manipulation it is straightforward to show that

$$\mathbf{A} = \frac{\mathbf{P}^T \mathbf{E}(D, Y) \mathbf{A}^-}{\mathbf{u}_2^T \mathbf{P}^T \mathbf{E}(D, Y) \mathbf{A}^-}. \quad (4)$$

Similarly, the backward message is a function of an input backward message  $\mathbf{B}$ , extrinsic information message  $D$ , and channel output  $Y$ . The SPA gives the relation

$$\mathbf{B} = \frac{\mathbf{E}(D, Y) \mathbf{P} \mathbf{B}^-}{\mathbf{u}_2^T \mathbf{E}(D, Y) \mathbf{P} \mathbf{B}^-} \quad (5)$$

noting that the denominator terms in (4) and (5) are both scalar. The denominator terms are normalization constants, which are optional for intermediate messages; they are used here to force  $\mathbf{u}_2^T \mathbf{A} = \mathbf{u}_2^T \mathbf{B} = 1$  (that is, the elements of  $\mathbf{A}$  and  $\mathbf{B}$  can be treated as probabilities).

The channel message  $C$  conveys information about the channel observations from the Markov subgraph to the LDPC subgraph. The message carrying the channel information is expressed as a log-likelihood ratio (as is the convention in LDPC decoding), so the channel message may be given as

$$C = \sigma(Y) \log \frac{(\mathbf{A}^-)^T (\mathbf{I} - \mathbf{N}) \mathbf{P} \mathbf{B}^-}{(\mathbf{A}^-)^T \mathbf{N} \mathbf{P} \mathbf{B}^-}. \quad (6)$$

The extrinsic message  $D$  is calculated in the same manner as messages from a symbol-variable node to a parity-check node, and is also passed as a log-likelihood ratio. However, all incident messages from parity-check nodes are included in the calculation, and the incident channel message is excluded.

Finally, an appropriate message-passing schedule must also be specified. Although there are many choices, the schedule which is most relevant to the remainder of this work is defined as follows:

**Message-Passing Schedule.** *Divide the set of nodes in the factor graph into the set of factor nodes (both parity-checks and channel factors), and the set of variable nodes (both symbol variables and channel states). A full iteration proceeds in two steps: first perform the SPA at each factor node (in any order), passing the appropriate messages to each attached variable node; then calculate the SPA at each variable node (in any order), passing the appropriate messages to each attached factor node.*

This is a simple extension of the customary message-passing schedule for LDPC decoding, and alternatives are available. For instance, one proposed schedule uses  $j$  iterations in the LDPC code for every iteration in the Markov chain [17]; our schedule is equivalent to this schedule for  $j = 1$ . We do not consider the question of optimal scheduling in this work.

### C. Conditions for Density Evolution

Two conditions are required of a decoder for the efficient application of DE analysis: the *independence assumption* and *symmetry*, each of which is outlined in the following. We do not consider the *concentration theorem* [5], since this theorem is not required to show that the symbol error probability becomes small for a regular (as opposed to an irregular) LDPC code. Only the

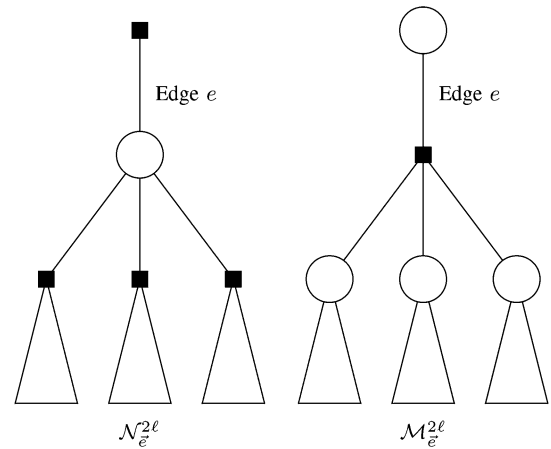


Fig. 3. The two types of directed local neighborhood,  $\mathcal{N}_e^{2\ell}$  and  $\mathcal{M}_e^{2\ell}$ . The defining difference is whether the message along  $e$  is passed from variable node to factor node or *vice versa*; the variable nodes or factor nodes could each be from either the GE or LDPC subgraph.

results from this section, and not the methods, are used in the rest of the paper.

**Cycle-Free Local Neighborhoods and the Independence Assumption.** We are interested in subgraphs of the GE-LDPC factor graph that contain only those nodes and edges involved in the calculation of a message along an edge  $e$  after a given number of iterations; we will refer to such a subgraph as the *directed local neighborhood of  $e$* , previously defined in [5], and use the message-passing schedule given in the previous section.

Formally, the directed local neighborhood  $\mathcal{N}_e^d$  of depth  $d$  for an edge  $e$  connecting a parity-check node  $h$  and a symbol-variable node  $X$  is defined as the graph including all nodes and edges traversed by all paths containing  $d$  edges and originating at variable node  $X$  such that  $e$  is not the first edge traversed by any such path. We will say that  $e$  is the *apex* of the local neighborhood.

We use the notation  $\mathcal{M}_e^d$  if the starting direction of the edge  $e$  is reversed, i.e.,  $\mathcal{M}_e^d$  is the set traversed by all paths of length  $d$  originating at parity-check node  $h$  such that  $e$  is not the first edge traversed. These two types of local neighborhoods are depicted in Fig. 3. From the message-passing schedule, it is easy to see that  $\mathcal{N}_e^{2\ell}$  is the set of nodes and edges involved in the calculation of the message passed from a variable node to a factor node along  $e$  after  $\ell$  iterations. (This includes messages passed in the GE Markov chain.)

To make analysis tractable, we require the messages passed within a local neighborhood to be independent, a condition which is called the *independence assumption*. A necessary condition for the independence assumption is that the directed local neighborhood of an edge be cycle free to some fixed depth  $2\ell$  with  $\text{Pr} \rightarrow 1$  as the block length  $n \rightarrow \infty$ . To show this, the stochastic properties of the LDPC subgraph must be defined. Let the ensemble of LDPC subgraphs be identical to the ensemble defined in [5], as follows. Let  $i$  be the number of *sockets* (i.e., connection points for an edge) in both the set of symbol-variable nodes and the set of parity-check nodes. Since the LDPC graph is bipartite, the number of sockets is the same in both sets. Label the symbol-variable and parity-check node sockets from 1 to  $i$ , and obtain a permutation  $\pi$  on  $\{1, \dots, i\}$ ,

where the permutation is selected at random with a uniform distribution over all such permutations. Finally, draw edges from symbol-variable socket  $j$  to parity-check socket  $\pi(j)$ .

Using this ensemble, it is known that the LDPC subgraph satisfies the cycle-free property, and the GE-LDPC graph can be shown to satisfy the property as well. Although this result and its proof are similar to the claim and proof, given in [5], that the LDPC local neighborhoods are cycle free, we include these results because of the importance of cycle-free local neighborhoods in our later exposition.

*Theorem 1:* For any fixed  $\ell$ ,  $\Pr[\mathcal{N}_e^{2\ell}$  is cycle free]  $\rightarrow 1$  as  $n \rightarrow \infty$ .

*Proof:* Here we present a sketch of the proof; a rigorous proof can be obtained by following the arguments from [5, Appendix A]. First, calculate

$$\Pr[\mathcal{N}_e^{2j} \text{ is cycle free} \mid \mathcal{N}_e^{2j-2} \text{ is cycle free}]$$

for some  $j \leq \ell$ . Let  $u_j$  and  $v_j$  represent the number of variable and factor nodes, respectively, in a local neighborhood  $\mathcal{N}_e^{2j}$ . A loop is formed: 1) if a parity-check node is selected twice, or 2) if the time index of a symbol-variable node is sufficiently close to the time index of an already selected node (because the Markov factor graphs will join as the graph is further constructed). Under the second condition, if a symbol variable is selected, then at most  $2\ell$  symbol variables are *forbidden* in order to prevent cycles. Following [5], it is straightforward to show that

$$\begin{aligned} \Pr[\mathcal{N}_e^{2j} \text{ is cycle free} \mid \mathcal{N}_e^{2j-2} \text{ is cycle free}] \\ \geq \left(1 - \frac{(d_c/d_v)v_\ell}{n}\right)^{v_j-v_{j-1}} \left(1 - \frac{2\ell u_\ell}{n}\right)^{u_j-u_{j-1}} \end{aligned}$$

where  $d_c$  is the check degree and  $d_v$  is the variable degree; and taking the product of these conditional densities from  $j = 1$  to  $\ell$

$$\Pr[\mathcal{N}_e^{2\ell} \text{ is cycle free}] \geq \left(1 - \frac{(d_c/d_v)v_\ell}{n}\right)^{v_\ell} \left(1 - \frac{2\ell u_\ell}{n}\right)^{u_\ell}.$$

Since  $u_\ell$  does not depend on  $n$ ,  $\Pr[\mathcal{N}_e^{2\ell}$  is cycle free]  $\rightarrow 1$  as  $n \rightarrow \infty$ .  $\square$

An example GE-LDPC local neighborhood is given in Fig. 4.

The cycle-free condition is necessary, but not sufficient, to prove the independence assumption, because the Markov memory of the channel states implies that these states can be dependent (however weakly) over long periods of time. In fact, if local neighborhood  $\mathcal{N}_e^{2\ell}$  is selected from the factor graph for a finite-length code, this means that the independence assumption cannot hold.

However, if the block length of the code approaches  $\infty$ , we may make the following argument. Let  $(X_1, \dots, X_n)$  be a codeword, and let  $d(X_i, X_j) := |i - j|$  be the difference in subscript between two codeword symbols  $X_i$  and  $X_j$ . In defining the channel, we assumed that symbols are transmitted in order of subscript, so  $d(X_i, X_j)$  measures the separation of two symbols in time. Suppose  $X_i$  is in a particular local neighborhood  $\mathcal{N}_e^{2\ell}$ . As a consequence of Theorem 1, the probability that no symbol-variable nodes  $X_j$  are selected such that  $d(X_i, X_j) <$

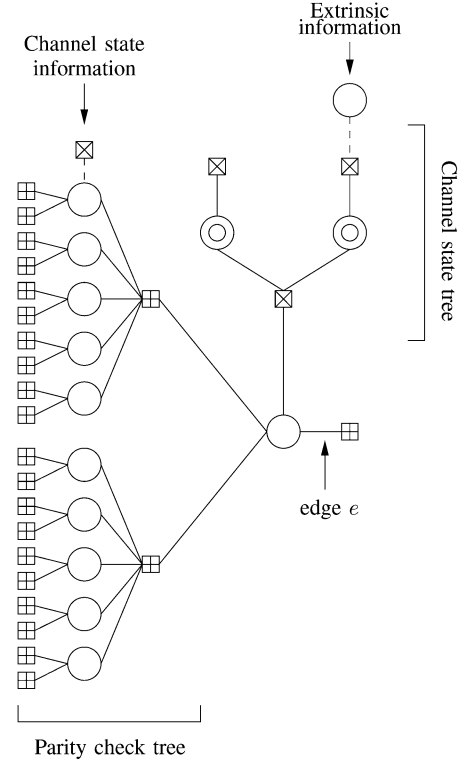


Fig. 4. A segment of the local neighborhood  $\mathcal{N}_e^{2\ell}$  of edge  $e$ .

$2\ell$  is bounded below by  $(1 - 2\ell u_\ell/n)^{u_\ell}$ , which approaches 0 as  $n \rightarrow \infty$ , since  $u_\ell$  is constant with respect to  $n$ . However, this is still true if the constant  $2\ell$  is replaced with some function of  $n$ , say  $f(n)$ , with the properties  $\lim_{n \rightarrow \infty} f(n) = \infty$  and  $\lim_{n \rightarrow \infty} f(n)/n = 0$ . An example of such a function is  $f(n) = \log n$ . Since the subscripts of symbol variables in the local neighborhood may therefore be arbitrarily separated with  $\Pr \rightarrow 1$ , the mutual information between their associated states goes to zero, so the state sequences contributing to the channel message at each symbol variable may be assumed to be independent.

**Symmetry.** Recall the definition of  $\sigma(\cdot)$  from the previous section. A channel message  $C$  is symmetric if it can be written  $C = \sigma(X)Q$ , where  $X$  is the symbol corresponding to the message  $C$ , and  $Q$  is a random variable independent of the transmitted codeword. If channel messages are symmetric then the performance of the decoder is independent of the transmitted codeword. This dramatically reduces the complexity of calculating DE, in that it allows us to perform the calculations with respect to a fixed codeword—in most cases, the all-zero codeword—with the guarantee that all codewords will perform identically. We need only establish the symmetry condition on the channel messages, since if these are symmetric, all messages passed within the LDPC subgraph are also symmetric [5].

We proceed in two steps: first, we show that  $C$  is symmetric if the forward and backward messages,  $\mathbf{A}$  and  $\mathbf{B}$ , are independent of the transmitted codeword; and second, since  $\mathbf{A}$  and  $\mathbf{B}$  are obtained from the extrinsic message  $D$ , we verify that  $\mathbf{A}$  and  $\mathbf{B}$  satisfy this independence as long as  $D$  is symmetric.

In the first step, we note that  $C$  is a function of the probability of the channel state  $p(s)$ , which is itself a function of  $\mathbf{A}$  and

**B.** Under the SPA,  $C$  is calculated as in (6), which can also be written

$$C = \log \frac{\sum_{s \in \mathcal{S}} p(y|x=0, s)p(s)}{\sum_{s \in \mathcal{S}} p(y|x=1, s)p(s)}.$$

However, since  $p(y|x, s)$  is the pmf of a binary-symmetric channel, it is easy to show that

$$C = \sigma(y) \log \frac{\sum_{s \in \mathcal{S}} p(y=0|x=0, s)p(s)}{\sum_{s \in \mathcal{S}} p(y=0|x=1, s)p(s)}. \quad (7)$$

Furthermore, there exists a random variable  $T \in \{0, 1\}$ , selected by the channel and independent of  $X$ , so that for given values  $Y = y, X = x$ , and  $T = t$ ,  $\sigma(y) = \sigma(x)\sigma(t)$ . Substituting this for  $\sigma(y)$  in (7), both  $\sigma(t)$  and everything under the logarithm are independent of the transmitted codeword (under the assumption that  $\mathbf{A}$  and  $\mathbf{B}$  are independent of the codeword), so  $C$  satisfies the symmetry condition.

The second step proceeds by induction. In the initial step of the induction, there is no input value of  $\mathbf{A}$ , so we use the prior density  $p(s)$  instead. A look at Fig. 2 shows that the SPA calculation of  $\mathbf{A}$  is a marginalization over  $S_i$  and  $X_i$ . (We will drop the subscripts for convenience, so this is a marginalization over  $S$  and  $X$ .) The marginalization can occur in either order, so we will consider  $X$  first. The marginalization over  $X$  for particular values of the channel observation  $Y = y$ , current state  $S = s$ , and future state  $S^+ = s^+$ , designated  $m(y, s, s^+)$ , is given in sum-product form by

$$m(y, s, s^+) = \sum_{x \in \{0, 1\}} p(s^+|s)p(s)p(y|x, s)p(x)$$

where  $p(x) = \gamma(D, x)$ . Expanding the expression for each value of  $y$ , and recalling that  $\eta_s$  is the inversion probability in state  $s$ , we may write

$$\begin{aligned} m(y, s, s^+) &= p(s^+|s)p(s) \\ &\cdot \begin{cases} \gamma(D, 0)(1 - \eta_s) + \gamma(D, 1)\eta_s, & y = 0 \\ \gamma(D, 0)\eta_s + \gamma(D, 1)(1 - \eta_s), & y = 1 \end{cases} \\ &= p(s^+|s)p(s)(\gamma(D, y)(1 - \eta_s) + \gamma(D, \bar{y})\eta_s) \end{aligned}$$

where  $\bar{y}$  represents the logical inverse of  $y$ . However, by assumption,  $D$  is symmetric and can be written  $D = \sigma(x)Q$ , where  $Q$  is a random variable independent of the transmitted codeword. Also, remember that  $\sigma(y) = \sigma(x)\sigma(t)$ . Making the substitution, we may write

$$m(y, s, s^+) = p(s^+|s)p(s)(\gamma(Q, t)(1 - \eta_s) + \gamma(Q, t)\eta_s)$$

which is independent of the transmitted codeword. The subsequent marginalization over  $S$ ,  $\sum_s m(y, s, s^+)$ , cannot introduce a new dependence on the codeword, so we can conclude that  $A$  is independent of the codeword as long as  $D$  is symmetric. In the induction step, if the input message  $\mathbf{A}$  is independent of the transmitted codeword, the only difference is that there is a different  $p(s)$ , which is still independent of the transmitted codeword. Thus, by induction, the messages  $\mathbf{A}$  are independent of the transmitted codeword. The same argument can be used for  $\mathbf{B}$ .

Thus, we may make the following statement.

*Theorem 2:* For SPA decoding in the GE channel, if all messages  $D$  satisfy the symmetry property, then all messages  $C$  satisfy the symmetry property, and so the probability of error for any symbol  $X$  is independent of the transmitted codeword.

*Proof:* This statement follows from the arguments given above, as well as the definition and proof of the symmetry condition given in [5].  $\square$

No equivalent ‘‘symmetry condition’’ exists for  $\mathbf{A}$  and  $\mathbf{B}$ , which are passed as functions of the channel state. The symmetry condition simplified DE in the LDPC subgraph by allowing DE to assume that every symbol variable was set to zero, so a lack of symmetry for messages as a function of  $S$  implies that each possible channel state sequence must be considered separately. Fortunately, this does not pose an undue burden. The channel state messages can be shown to form a Markov process, so the complexity scales with the size of the state space, which is small. This property follows from the calculation of the channel state messages incident to a channel factor node. In the derivation of the DE algorithm, an intermediate step in the calculation of each message is the conditioning of that message on both possible values of the state, which also simplifies the subsequent marginalization operations.

### III. ANALYSIS OF GE-LDPC DECODING

In Section II, we established that the requirements of density evolution are satisfied by GE-LDPC decoders. In this section, we present the main contribution of our work: characterization of a three-dimensional decoding region induced by points in the GE parameter space where successful decoding occurs. (Since the GE parameter space is four-dimensional, ideally we would like a four-dimensional characterization; a result for the remaining dimension is a subject of active research.)

We introduce this section with some notation. Let  $P_{\text{err}}[\ell]$  represent the probability of symbol error for the factor-graph decoder represented by the cycle-free local neighborhood  $\mathcal{N}_e^{2\ell}$ , where  $e$  is an edge from a symbol-variable node to a parity-check node (that is, for a sufficiently long code,  $P_{\text{err}}[\ell]$  is the *a posteriori* probability of symbol error after the  $\ell$ th iteration of GE-LDPC decoding). To designate this quantity for a specific point  $c := (b, g, \eta_B, \eta_G)$  in the GE parameter space, we write  $P_{\text{err}}[\ell](b, g, \eta_B, \eta_G)$  or  $P_{\text{err}}[\ell](c)$ .

#### A. Side Information

Analysis of decoders using DE depends only on the structure of the local neighborhood  $\mathcal{N}_e^{2\ell}$ , which is a subgraph of the overall GE-LDPC factor graph. Furthermore, using the subset  $\mathbf{Y}_e$  of observations associated with this subgraph,  $e$  connects a symbol variable node to a parity-check node. The SPA and DE both assume that, if the local neighborhood is cycle free, then the pmf of  $\mathbf{Y}_e$ , along with the appropriate hidden variables, is completely described by the local neighborhood; and therefore, the SPA calculates  $p(x|\mathbf{y}_e)$ . We will not prove that the SPA actually calculates  $p(x|\mathbf{y}_e)$  in a general setting (though we conjecture that it is so, under conditions that hold with  $\text{Pr} \rightarrow 1$ ). However, under the independence assumption and the symmetry condition, we need only examine the factor graph  $\mathcal{N}_e^{2\ell}$  to obtain the

probability of error for decoding at the  $\ell$ th iteration of the SPA. Furthermore, since this factor graph is cycle free, under the SPA, it implements an optimal *a posteriori* detector for  $X$  given  $\mathbf{Y}_e$ .

Our general approach in the sequel will be to take two GE channels,  $c$  and  $c^*$ , with different parameters, and show that the local neighborhood for  $c$  can be constructed from the local neighborhood for  $c^*$  and the factor graph representing some side information  $H$ . Since the cycle-free local neighborhood  $\mathcal{N}_e^{2\ell}$  represents an optimal decoder up to depth  $2\ell$  for the symbol  $X$ , the deletion of  $H$  cannot improve the probability of error, so we can order the probabilities of error under  $c$  and  $c^*$ . We formalize the consequences of deleting side information with the following straightforward lemma.

*Lemma 1 (Side Information):* Let  $H \in \mathcal{H}$  and  $Y \in \mathcal{Y}$  be the outputs of two independent channels with common input  $X \in \{0, 1\}$ . Let  $d_X(H, Y)$  denote the detection function that minimizes probability of error, i.e., that minimizes

$$P_e(d_X(H, Y)) := \Pr[X \neq d_X(H, Y)]$$

when both channel outputs are observed, and let  $d_X(Y)$  denote the detection function that minimizes probability of error when only  $Y$  is observed. Then  $P_e(d_X(H, Y)) \leq P_e(d_X(Y))$ .

*Proof:* The key to the proof is that  $d_X(H, Y)$  always has the option to throw out  $H$ . Define  $\hat{d}_X(H, Y) : \mathcal{H} \times \mathcal{Y} \rightarrow \{0, 1\}$  so that  $\hat{d}_X(H, Y) := d_X(Y)$  for all  $H$  and  $Y$ , meaning that  $\hat{d}_X(H, Y)$  is only trivially (or formally) a function of  $H$ . Since  $d_X(H, Y)$  is the detector which minimizes  $P_e$  as a function of  $H$  and  $Y$ , then obviously

$$P_e(d_X(H, Y)) \leq P_e(\hat{d}_X(H, Y)) = P_e(d_X(Y)). \quad \square$$

Lemma 1 is a generalization of a technique used in the Proof of Theorem 1 in [5], which proved monotonicity for physically degraded channels.

As our first result from Lemma 1, and to give a flavor of its use, we give the following useful results that  $P_{\text{err}}[\ell]$  is monotonically nonincreasing with  $\ell$ , and that its limit exists, generalizing the result from [3, Theorem 7]) to the GE-LDPC case.

*Theorem 3:* For any integer  $\ell \geq 1$  and any GE channel  $c \in \mathcal{G}$ :

- $P_{\text{err}}[\ell](c) \geq P_{\text{err}}[\ell + 1](c)$  and
- $\lim_{\ell \rightarrow \infty} P_{\text{err}}[\ell](c)$  exists (and is finite).

*Proof:* To prove the first statement, let  $\mathbf{Y}_e$  represent the observations in  $\mathcal{N}_e^{2\ell}$ , and let  $\mathbf{Y}_e^+$  represent the observations in  $\mathcal{N}_e^{2(\ell+1)}$ , both of which are assumed to be cycle free. Note that  $P_{\text{err}}[\ell](c)$  is the probability of error for the optimal decoder for  $\mathbf{Y}_e$ . Furthermore, under iterative decoding, clearly all the evidence  $\mathbf{Y}_e$  is included in  $\mathbf{Y}_e^+$ . That is, if an observation is used by the decoder after  $\ell$  iterations of decoding, it is also used after  $\ell + 1$  iterations of decoding. Since  $P_{\text{err}}[\ell + 1](c)$  is the probability of error for the optimal decoder for  $\mathbf{Y}_e^+$ , which contains  $\mathbf{Y}_e$  and possibly some extra observations, discarding those extra observations cannot decrease the probability of error, so  $P_{\text{err}}[\ell](c) \geq P_{\text{err}}[\ell + 1](c)$ .

To prove the second statement, the sequence  $P_{\text{err}}[\ell](c)$  is bounded between 0 and 1 for all  $\ell$  (by the definition of probability), and is nonincreasing in  $\ell$  (by the first statement). Thus, a finite limit must exist.  $\square$

Practically, when an LDPC code over a given channel  $c$  is analyzed with DE, the result may be that  $P_{\text{err}}[\ell](c)$  approaches zero to within the precision of a computer representation. A system designer is normally only interested in whether the probability of error is at or below some very small but acceptable value (say,  $10^{-8}$ , which is comparable to the limits of machine precision). Furthermore, a system designer is normally constrained by a maximum number of iterations, designated  $\ell_{\text{max}}$ . Letting  $\epsilon$  be the maximum acceptable error probability, we will be interested in the cases where  $\lim_{\ell \rightarrow \infty} P_{\text{err}}[\ell](c) < \epsilon$ , which will correspond to the case of successful decoding, and from Theorem 3, we know that  $P_{\text{err}}[\ell](c) \leq \epsilon$  for some  $\ell \leq \ell_{\text{max}}$  implies that both  $P_{\text{err}}[\ell_{\text{max}}](c) \leq \epsilon$  and  $\lim_{\ell \rightarrow \infty} P_{\text{err}}[\ell](c) \leq \epsilon$ , which guarantees successful decoding. We will also be interested in cases where  $P_{\text{err}}[\ell_{\text{max}}](c) > \epsilon$ . The set of points  $c \in \mathcal{G}$  where  $P_{\text{err}}[\ell_{\text{max}}](c) \leq \epsilon$  will be referred to as the *decoding region* (i.e., the region of parameters where the decoding is successful), while the set of points where  $P_{\text{err}}[\ell_{\text{max}}](c) > \epsilon$  will be referred to as the *nondecoding region*.

## B. BSC Concatenation

Our first result arises from the observation that a GE channel concatenated with a BSC results in a different GE channel. Consider a GE channel with parameters  $(b, g, \eta_B, \eta_G) \in \mathcal{G}$ , whose output is passed through a BSC with parameter  $\epsilon$ . Since the BSC is independent of the channel state sequence, the state transition probabilities are unchanged, so from the above, the output is a GE channel with different inversion probabilities.

Intuitively, concatenating a GE channel with a BSC should increase the probability of error. For a given channel  $c \in \mathcal{G}$ , there will be a set of channels in  $\mathcal{G}$  that result from the concatenation of  $c$  with a BSC; there will also be a set of channels in  $\mathcal{G}$  that, when concatenated with a particular BSC, result in  $c$ . We will argue that any channel in the former set has higher  $P_{\text{err}}$  than  $c$  under LDPC decoding, while any channel in the latter set has lower  $P_{\text{err}}$  than  $c$ .

Formally, define the function  $b : \mathcal{G} \times [0, 1/2] \rightarrow \mathcal{G}$  so that  $b(c, \alpha)$  returns the channel that results from concatenating  $c$  with a BSC having parameter  $\alpha$ , where  $0 \leq \alpha \leq 1/2$ . The set  $\mathcal{A}^{(c)}$  (the *former* set from the intuitive discussion) is defined as

$$\mathcal{A}^{(c)} := \{u : u \in \mathcal{G}, \exists \alpha \text{ s.t. } b(c, \alpha) = u\}$$

while the set  $\mathcal{B}^{(c)}$  (the *latter* set from the intuitive discussion) is defined as

$$\mathcal{B}^{(c)} := \{u : u \in \mathcal{G}, \exists \alpha \text{ s.t. } b(u, \alpha) = c\}.$$

The following lemma relates  $c$  geometrically to the channels in  $\mathcal{A}^{(c)}$  and  $\mathcal{B}^{(c)}$ .

*Lemma 2:* Let  $c = (b, g, \eta_B, \eta_G) \in \mathcal{G}$  represent a GE channel, and let  $\mathcal{A}^{(c)}$  and  $\mathcal{B}^{(c)}$  be defined as above. Then we have the following:

- for constant  $(b, g)$ ,  $\mathcal{A}^{(c)}$  consists of all  $u = (b, g, \eta_B^*, \eta_G^*)$  satisfying

$$\eta_G^* = \eta_B^* \frac{1 - 2\eta_G}{1 - 2\eta_B} - \frac{\eta_B - \eta_G}{1 - 2\eta_B} \quad (8)$$

where  $\eta_B^* \geq \eta_B$  and  $\eta_G^* \geq \eta_G$ , and

- for constant  $(b, g)$ ,  $\mathcal{B}^{(c)}$  consists of all  $u = (b, g, \eta_B^*, \eta_G^*)$  satisfying (8), where  $\eta_B^* \leq \eta_B$  and  $\eta_G^* \leq \eta_G$ .

*Proof:* Consider the first part of the lemma. If  $u$  is formed by concatenating  $c$  with a BSC having parameter  $\alpha$ , then

$$\eta_B^* = \eta_B(1 - \alpha) + (1 - \eta_B)\alpha$$

and

$$\eta_G^* = \eta_G(1 - \alpha) + (1 - \eta_G)\alpha.$$

Solving both equations for  $\alpha$ , and a minor manipulation, lead to (8). The first part then follows by observing that  $\alpha \in [0, 1/2]$  implies that  $\eta_B^* \geq \eta_B$  and  $\eta_G^* \geq \eta_G$ .

In the second part of the lemma,  $c$  is formed by concatenating  $u$  with a BSC having parameter  $\alpha$ . Now

$$\eta_B = \eta_B^*(1 - \alpha) + (1 - \eta_B^*)\alpha$$

and

$$\eta_G = \eta_G^*(1 - \alpha) + (1 - \eta_G^*)\alpha.$$

Again, solving for  $\alpha$  leads to (8), though in this case,  $\alpha \in [0, 1/2]$  implies that  $\eta_B^* \leq \eta_B$  and  $\eta_G^* \leq \eta_G$ .  $\square$

The importance of Lemma 2 is to give a geometric interpretation to  $\mathcal{A}^{(c)}$  and  $\mathcal{B}^{(c)}$ . If  $b$  and  $g$  are fixed, the set of possible values of  $\eta_B$  and  $\eta_G$  can be represented on a plane, and the line segments representing  $\mathcal{A}^{(c)}$  and  $\mathcal{B}^{(c)}$  can be plotted on this plane using (8). The line represented by (8) passes through  $c$ , and the segments of this line described in each part of the lemma terminate at  $c$ . For ease of visualization, it is easy to show that this line also passes through  $(\eta_B^*, \eta_G^*) = (1/2, 1/2)$  for any  $c$ . These results can be easily generalized to the case where  $1/2 \leq \alpha \leq 1$ .

Side information consisting of the BSC noise sequence allows the original channel to be restored. This observation leads directly to the following theorem.

*Theorem 4 (BSC Concatenation):* Let  $(b, g, \eta_B, \eta_G)$  be a point in  $\mathcal{G}$  for which  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ . Then all points  $(b, g, \eta_B^*, \eta_G^*)$  along the line segment satisfying (8), where  $\eta_B^* \leq \eta_B$  and  $\eta_G^* \leq \eta_G$ , also converge to  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ .

*Proof:* If there exist pairs  $(\eta_B, \eta_G)$  and  $(\eta_B^*, \eta_G^*)$  such that  $\eta_B^* \leq \eta_B$  and  $\eta_G^* \leq \eta_G$ , and lying along the given line, then from Lemma 2 there exists a BSC with parameter  $\xi \geq 0$  such that a GE channel with inversion probabilities  $(\eta_B^*, \eta_G^*)$  concatenated with that BSC has inversion probabilities  $(\eta_B, \eta_G)$ . Let  $\mathbf{Z}_{\text{BSC}}$  be the noise sequence of this BSC. Furthermore, let  $\mathbf{Y}$  and  $\mathbf{Y}^*$  be the channel outputs for the channel with inversion probabilities  $(\eta_B, \eta_G)$  and  $(\eta_B^*, \eta_G^*)$ , respectively, where  $\mathbf{Y} = \mathbf{Y}^* \oplus \mathbf{Z}_{\text{BSC}}$ . Knowing  $\mathbf{Y}$  and  $\mathbf{Z}_{\text{BSC}}$ ,  $\mathbf{Y}^*$  can be easily recovered. Since  $\mathbf{Z}_{\text{BSC}}$  is independent of  $\mathbf{Y}^*$ , an optimal decoder with knowledge of  $\mathbf{Z}_{\text{BSC}}$  and  $\mathbf{Y}^*$  is equivalent to an optimal decoder with knowledge of  $\mathbf{Y}$ . Thus, an optimal decoder for  $\mathbf{Y}$  with side knowledge of  $\mathbf{Z}_{\text{BSC}}$  is equivalent to an optimal decoder for  $\mathbf{Y}^*$ . From the side information lemma, this is sufficient to show that

$$P_{\text{err}}[\ell](b, g, \eta_B^*, \eta_G^*) \leq P_{\text{err}}[\ell](b, g, \eta_B, \eta_G)$$

and, therefore, that if  $P_{\text{err}}[\ell_{\text{max}}](b, g, \eta_B, \eta_G) \leq \epsilon$ , then  $P_{\text{err}}[\ell_{\text{max}}](b, g, \eta_B^*, \eta_G^*) \leq \epsilon$  as well.  $\square$

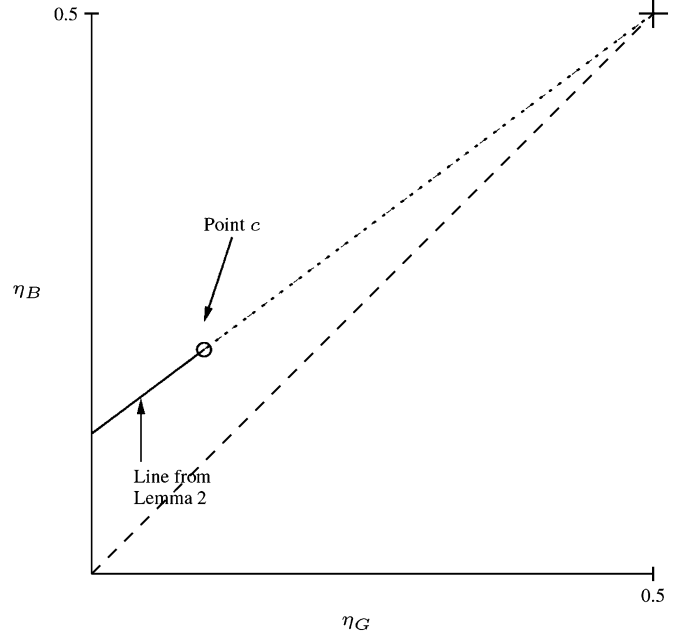


Fig. 5. One-dimensional decoding region, showing a region formed as a result of applying Theorem 4 to a point which converges to  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ . All points along the solid line also have  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ .

One may also say that

$$P_{\text{err}}[\ell](b, g, \eta_B^*, \eta_G^*) \leq P_{\text{err}}[\ell](b, g, \eta_B, \eta_G)$$

because the channel  $(b, g, \eta_B, \eta_G)$  is physically degraded with respect to  $(b, g, \eta_B^*, \eta_G^*)$ . The results in Theorem 4 are depicted in Fig. 5.

The converse of the theorem may be stated briefly as follows: if  $P_{\text{err}}[\ell_{\text{max}}](c) > \epsilon$  for some point  $c = (b, g, \eta_B, \eta_G)$ , then any point  $(b, g, \eta_B^*, \eta_G^*)$  cannot have better  $P_{\text{err}}[\ell_{\text{max}}]$  if that point was formed by concatenating  $(b, g, \eta_B, \eta_G)$  with a BSC. Using the first part of Lemma 2, we have the following corollary.

*Corollary:* Let  $(b, g, \eta_B, \eta_G)$  be a point for which  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ . Then  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$  for all points  $(b, g, \eta_B^*, \eta_G^*)$  along the line segment satisfying (8) where  $\eta_B^* \geq \eta_B$  and  $\eta_G^* \geq \eta_G$ , and for which  $(b, g)$  are kept constant.

### C. State Scrambling

Our second result implies that for constant  $\bar{\eta}$ , as defined in (1), increasing the difference between  $\eta_B$  and  $\eta_G$  leads to a decrease in  $P_{\text{err}}[\ell]$ . In other words, we are proving the intuition that, all else being equal, a larger contrast between  $\eta_B$  and  $\eta_G$  is better than a smaller contrast, because the states are easier to distinguish. To show this, we introduce a proof technique called *state scrambling*. Intuitively, state scrambling acts like a partial interleaver, which selects a subset of the channel outputs and subjects only that subset to a random permutation, while leaving the rest alone. Ideally, for the time instants within the subset, the corresponding channel states would be completely randomized, independent of each other and the states in the complementary, noninterleaved subset. The decoder could then be provided with side information consisting of the identity of this subset and the permutation used by the interleaver, which would allow the operation to be inverted. However, to avoid difficult arguments related to perfect

interleaving, we simply replace the channel state at randomly selected instants with an independently selected state.

Consider a GE-like channel with a state sequence generated in the following manner. Let  $\mathbf{S}$  be a state sequence with state transition probabilities  $(b, g)$ , generated as usual for a GE channel. Let  $p(s_k)$  represent the marginal pmf for the  $k$ th element of  $\mathbf{S}$ . Let  $\mathbf{S}'$  be a scrambled state sequence with independently distributed elements, i.e.,

$$p(\mathbf{s}') = \prod_{k=1}^n p(s'_k)$$

where  $p_{S'_k}(G) = p_{S_k}(G)$  and  $p_{S'_k}(B) = p_{S_k}(B)$  (i.e., the states are independent but the marginal state probabilities are the same). Let  $\mathbf{R} \in \{0, 1\}^n$  be a random *scrambling sequence* with independent Bernoulli-distributed elements, where  $\Pr(R_i = 1) = \psi$ . The resulting channel state sequence  $\hat{\mathbf{S}}$  is generated from  $\mathbf{R}, \mathbf{S}$ , and  $\mathbf{S}'$  through the relation

$$\hat{S}_k = \begin{cases} S_k, & R_k = 0 \\ S'_k, & R_k = 1. \end{cases}$$

For a channel  $c = (b, g, \eta_B, \eta_G)$ , if a state sequence  $\hat{S}_k$  generated in this way, where the state transition probabilities  $(b, g)$  are used to obtain  $\mathbf{S}$  and  $\mathbf{S}'$  (i.e.,  $p_{S'_k}(G) = g/(b+g)$  and  $p_{S'_k}(B) = b/(b+g)$ ), we say that the state scrambler with parameter  $\psi$  is applied to this channel  $c$ .

We will show that applying state scrambling to a GE channel results in a different GE channel. Our argument here is similar to our argument in the case of BSC concatenation: given a GE channel  $c$ , there are channels which can be generated by applying state scrambling to  $c$ , and channels for which  $c$  can be generated by applying state scrambling with a particular parameter value. For some  $\psi \in [0, 1]$ , let  $r : \mathcal{G} \times [0, 1] \rightarrow \mathcal{G}$  be the function so that  $r(c, \psi)$  returns the channel that results from applying state scrambling with parameter  $\psi$  to  $c$ . Similarly to the previous section, define

$$\mathcal{C}^{(c)} := \{u : u \in \mathcal{G}, \exists \psi \text{ s.t. } r(c, \psi) = u\}$$

and

$$\mathcal{D}^{(c)} := \{u : u \in \mathcal{G}, \exists \psi \text{ s.t. } r(u, \psi) = c\}.$$

The sets  $\mathcal{C}^{(c)}$  and  $\mathcal{D}^{(c)}$  are geometrically related to  $c$  through the following lemma.

*Lemma 3:* Let  $c = (b, g, \eta_B, \eta_G) \in \mathcal{G}$  represent a GE channel, let  $\bar{\eta} = (b\eta_B + g\eta_G)/(b+g)$ , and let  $\mathcal{C}^{(c)}$  and  $\mathcal{D}^{(c)}$  be defined as above. Then we have the following.

- If  $u = r(c, \psi)$ , then

$$u = (b, g, \eta_B(1-\psi) + \bar{\eta}\psi, \eta_G(1-\psi) + \bar{\eta}\psi).$$

- For constant  $(b, g)$ ,  $\mathcal{C}^{(c)}$  consists of all  $u = (b, g, \eta_B^*, \eta_G^*)$  satisfying

$$\eta_G^* = \eta_B^* \frac{\bar{\eta} - \eta_G}{\bar{\eta} - \eta_B} - \frac{\bar{\eta}(\eta_B - \eta_G)}{\bar{\eta} - \eta_B} \quad (9)$$

and where  $\eta_B - \eta_G \geq \eta_B^* - \eta_G^*$ .

- For constant  $(b, g)$ ,  $\mathcal{D}^{(c)}$  consists of all  $u = (b, g, \eta_B^*, \eta_G^*)$  satisfying (9), and where  $\eta_B - \eta_G \leq \eta_B^* - \eta_G^*$ .

*Proof:* To show the first part of the lemma, we show that any noise sequence generated by the scrambler has the same probability as the given GE channel. To calculate the probability of a noise sequence  $\mathbf{Z}$  generated by the state scrambler, we may write  $p(\mathbf{z}) = \sum_{\mathbf{s}} p(\mathbf{z}|\mathbf{s})p(\mathbf{s})$ , where

$$p(\mathbf{z}|\mathbf{s}) = \prod_i \sum_{r_i, s'_i} p(z_i | s_i, r_i, s'_i) p(r_i) p(s'_i)$$

and  $p(\mathbf{s})$  is unchanged from the regular GE case. The inner term  $p(z_i | s_i, r_i, s'_i)$  arises from the fact that if  $R_i = 0$ ,  $Z_i$  is dependent on  $S_i$ , while if  $R_i = 1$ ,  $Z_i$  is dependent on the independently selected state  $S'_i$ . Evaluating these expressions, the inner sum reduces to  $\prod_i p'(z_i | s_i)$ , where

$$p'(1|G) = \eta_G(1-\psi) + \bar{\eta}\psi$$

and

$$p'(1|B) = \eta_B(1-\psi) + \bar{\eta}\psi$$

which is equivalent to a GE channel with the given inversion probabilities. The second and third parts of the lemma follow straightforwardly from manipulations similar to those used in Lemma 2.  $\square$

Again, a line is described by the condition in (9), divided into two segments by the point  $(b, g, \eta_B, \eta_G)$ .

As we argued intuitively at the beginning of the section, the effect of state scrambling can be reversed using side information. We justify this statement in the following theorem.

*Theorem 5 (State Scrambling):* Let  $(b, g, \eta_B, \eta_G)$  be a point in  $\mathcal{G}$  such that  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ . Then all points  $(b, g, \eta_B^*, \eta_G^*)$  in  $\mathcal{G}$  having the same  $(b, g)$  and satisfying (9), with  $\eta_B - \eta_G \leq \eta_B^* - \eta_G^*$ , also have  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ .

*Proof:* Using the result in Lemma 3, channels representing all points  $(b, g, \eta_B^*, \eta_G^*) \in \mathcal{G}$  along the given line segment may be applied to a state scrambler to generate  $(b, g, \eta_B, \eta_G)$ . The remainder of the proof is in the Appendix, part A.  $\square$

Again, the converse of this result follows similarly to the BSC concatenation theorem:

*Corollary:* Let  $(b, g, \eta_B, \eta_G)$  be a point in  $\mathcal{G}$  such that  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ . Then all points  $(b, g, \eta_B^*, \eta_G^*)$  along the line segment satisfying (9), where  $\eta_B - \eta_G \geq \eta_B^* - \eta_G^*$  for  $0 \leq \psi \leq 1$  and the same  $(b, g)$ , also have the property that  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ .

For fixed  $(b, g)$ , the results in Theorems 4 and 5 operate on the plane formed by pairs of inversion probabilities  $(\eta_B, \eta_G)$ . Since the result from Theorem 4 implies a one-dimensional decoding region on this plane that is not (in general) collinear with the one-dimensional decoding region implied by the state scrambling theorem, these two results taken together imply a two-dimensional decoding region, as given in Fig. 6. For some point  $(\eta_B, \eta_G)$  on this plane for which  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ , any point within the two-dimensional decoding region may be reached by combining moves provided by each theorem. Similarly, by combining the converses of these theorems, a two-dimensional nondecoding region is induced by each point for which  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ .

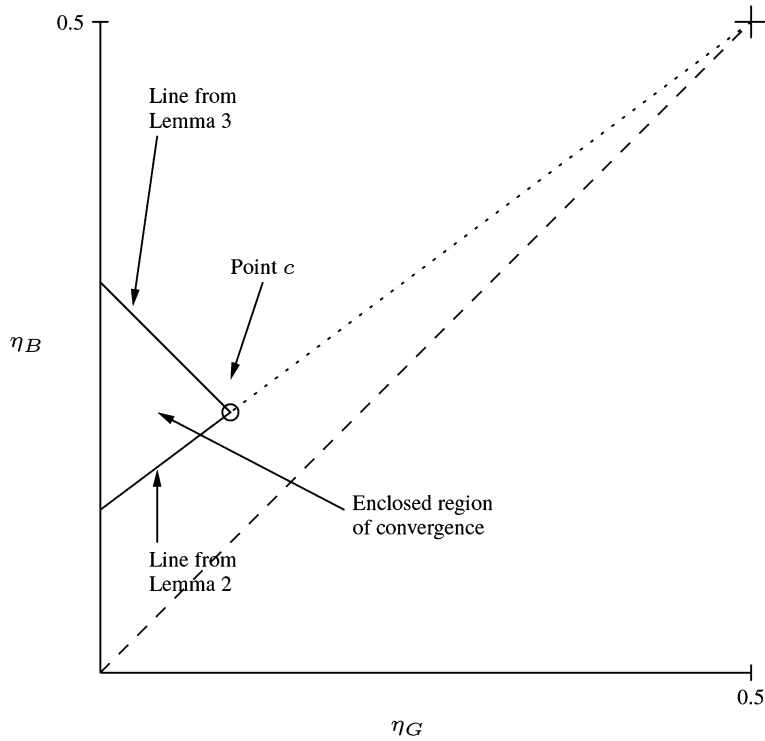


Fig. 6. Two-dimensional decoding region, showing a region formed as a result of applying Theorems 4 and 5 simultaneously to a point which converges to  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ . All points enclosed by the solid lines also have  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ .

*D. Segmentation*

Here we show that increasing the length of the channel memory decreases  $P_{\text{err}}[\ell]$ . We make use of a proof technique, which we call *segmentation*, similar to a method used in the proof of [19, Proposition 5].

Consider a set  $\Xi = \{\mathbf{S}^{(i)}\}_{i=1}^{\infty}$  of state sequences, where the sequences  $\mathbf{S}^{(i)}$  are independent and identically distributed, and all have the same state transition probabilities  $(b, g)$ . From  $\Xi$ , we construct a state sequence  $\tilde{\mathbf{S}}$  as follows. Let  $\{W_i\}_{i=1}^n$  be a random sequence of nondecreasing integers such that  $W_1 = 1$ , and for all  $i$ ,  $W_{i+1} = W_i + U_i$ , where  $U_i \in \{0, 1\}$  is an independent Bernoulli random variable for each  $i$ , with  $\text{Pr}(U_i = 1) = p$ . For example, a valid sequence of  $\{W_i\}_{i=1}^n$  would be  $\{1, 1, 1, 1, 2, 2, 3, 3, 3, 4 \dots\}$ . Given  $\{W_i\}_{i=1}^n$ , the elements of the sequence  $\tilde{\mathbf{S}}$  are defined as  $\tilde{S}_i = S_i^{(\omega_i)}$ . In other words, the state sequence  $\tilde{\mathbf{S}}$  is formed from segments of each independent state sequence  $\mathbf{S}^{(i)}$ . Again, for some point  $(b, g, \eta_B, \eta_G) \in \mathcal{G}$ , we say that segmentation with parameter  $p$  is applied to this channel if  $\tilde{\mathbf{S}}$  is generated such that the state transition probabilities  $(b, g)$  are used to generate each  $\mathbf{S}^{(i)}$  in  $\Xi$ .

The effect of segmentation is to introduce additional state transitions to an existing state sequence, and we will show that a noise sequence arising from segmentation is equivalent to a noise sequence arising from a GE channel. As we have argued in the cases of BSC concatenation and state scrambling, given a GE channel  $c$ , there are channels which can be generated by applying segmentation to  $c$ , and channels for which segmentation applied with a particular parameter  $p$  results in  $c$ . For some  $p \in [0, 1]$ , let  $q : \mathcal{G} \times [0, 1] \rightarrow \mathcal{G}$  be the function so that  $q(c, p)$

returns the channel that arises from applying segmentation with parameter  $p$  to  $c$ . Also, let

$$\mathcal{E}^{(c)} := \{u : u \in \mathcal{G}, \exists p \text{ s.t. } q(c, p) = u\}$$

and

$$\mathcal{F}^{(c)} := \{u : u \in \mathcal{G}, \exists p \text{ s.t. } q(u, p) = c\}.$$

The sets  $\mathcal{E}^{(c)}$  and  $\mathcal{F}^{(c)}$  are geometrically related to  $c$  through the following lemma.

*Lemma 4:* Let  $c^* = (b, g, \eta_B, \eta_G) \in \mathcal{G}$  represent a GE channel, let  $k = b/(b + g)$ , and let  $\mathcal{E}^{(c)}$  and  $\mathcal{F}^{(c)}$  be defined as above. Then we have the following.

- If  $u = s(c, p)$ , then

$$u = ((1 - p)b + pk, (1 - p)g + p(1 - k), \eta_B, \eta_G).$$

- For constant  $(\eta_B, \eta_G)$ ,  $\mathcal{E}^{(c)}$  consists of all  $u = (b^*, g^*, \eta_B, \eta_G)$  satisfying

$$b^* = g^* \frac{b}{g} \tag{10}$$

so that  $b^* + g^* \geq b + g$ .

- For constant  $(\eta_B, \eta_G)$ ,  $\mathcal{F}^{(c)}$  consists of all  $u = (b^*, g^*, \eta_B, \eta_G)$  satisfying (10), so that  $b^* + g^* \leq b + g$ .

*Proof:* To show the first part of the lemma, consider the state transition probability pair  $(b, g)$  that corresponds to the pair  $(b, g)$ . To show that  $\tilde{\mathbf{S}}$  is a GE state sequence, we may consider the sequence of increments  $\mathbf{U} \in \{0, 1\}^n$ , with

independent elements, such that  $\Pr(u_i = 1) = p$ . Since the sequences in  $\Xi$  are independent and identically distributed, we may write

$$p(\tilde{\mathbf{s}}) = p(\tilde{s}_1) \prod_{i=1}^{n-1} \sum_{u_i} p(\tilde{s}_{i+1} | \tilde{s}_i, u_i) p(u_i)$$

where  $p(\tilde{s}_{i+1} | \tilde{s}_i, u_i = 0) = p(s_{i+1} | s_i)$ , the state transition probability given by the original parameters  $(b, g)$ , and where  $p(\tilde{s}_{i+1} | \tilde{s}_i, u_i = 1) = p(s_{i+1})$ , the marginal state probabilities for  $S_{i+1}$ . Thus, we may equivalently write

$$p(\tilde{\mathbf{s}}) = p(s_1) \prod_{i=1}^{n-1} p'(s_{i+1} | s_i)$$

where

$$p'(\tilde{s}_{i+1} | \tilde{s}_i) = \begin{cases} 1 - (1-p)g - pg/(b+g), & \tilde{s}_i = B, & \tilde{s}_{i+1} = B \\ (1-p)g + pg/(b+g), & \tilde{s}_i = B, & \tilde{s}_{i+1} = G \\ (1-p)b + pb/(b+g), & \tilde{s}_i = G, & \tilde{s}_{i+1} = B \\ 1 - (1-p)b - pb/(b+g), & \tilde{s}_i = G, & \tilde{s}_{i+1} = G \end{cases}$$

and thus the state sequence  $\tilde{\mathbf{s}}$  is a GE state sequence with new state transition probability pair

$$(b^*, g^*) = ((1-p)b + pb/(b+g), (1-p)g + pg/(b+g)).$$

Defining an inversion probability function  $p(z_i | \tilde{s}_i)$  with an appropriate pair of inversion probabilities  $(\eta_B, \eta_G)$  results in a GE channel, which proves the first part of the lemma.

The second and third parts of the lemma follow from straightforward manipulations, analogous to Lemmas 2 and 3.  $\square$

Note that the condition in (10) forms a line, partitioned into two segments by  $(b, g, \eta_B, \eta_G)$ .

Once again, we apply the side information lemma to obtain the following theorem.

*Theorem 6 (Segmentation):* Let  $(b, g, \eta_B, \eta_G)$  be a point in  $\mathcal{G}$  such that  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ . Then all points  $(b^*, g^*, \eta_B, \eta_G)$  in  $\mathcal{G}$  having the same  $(\eta_B, \eta_G)$  and satisfying (10), with  $b^* + g^* \leq b + g$ , also have  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ .

*Proof:* Using the result in Lemma 4, if  $(b, g)$  satisfies (10), with the same  $(\eta_B, \eta_G)$  and  $b^* + g^* \leq b + g$ , then segmentation may be used to obtain the channel corresponding to the point  $(b, g, \eta_B, \eta_G)$  from the point  $(b^*, g^*, \eta_B, \eta_G)$ . The remainder of the proof is in the Appendix, part B.  $\square$

Again, the converse straightforwardly follows.

*Corollary:* Let  $(b, g, \eta_B, \eta_G)$  be a point in  $\mathcal{G}$  such that  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ . Then all points  $(b^*, g^*, \eta_B, \eta_G)$  along the line segment satisfying (10), for constant  $(\eta_B, \eta_G)$  and  $b^* + g^* \geq b + g$ , also have the property that  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ .

To visualize this result in three dimensions, for some point  $(b, g, \eta_B, \eta_G) \in \mathcal{G}$ , define a quantity  $\mu := 1 - b - g$ , which was introduced in [19], and measures the persistence of the channel

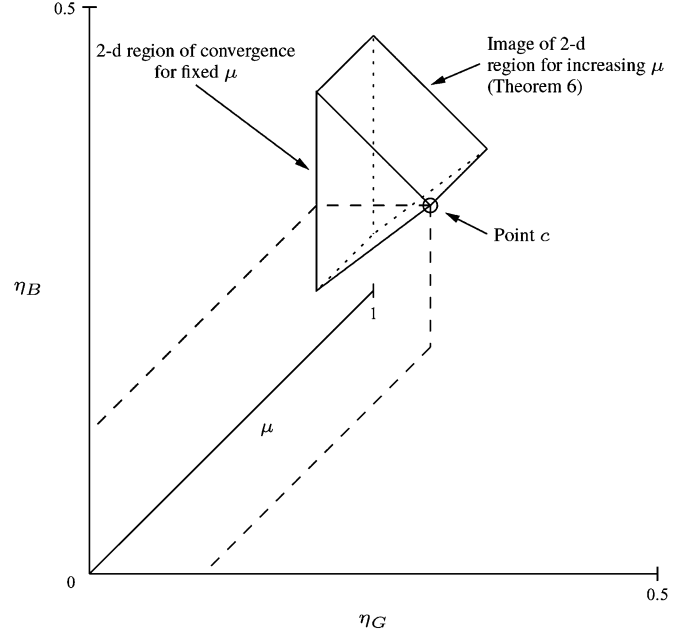


Fig. 7. Three-dimensional decoding region. To visualize a three-dimensional region in four-dimensional space, we define  $\mu := 1 - b - g$ , as in Section III-D. The two-dimensional region of Fig. 6 for a given point which converges to  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$  is extended along the  $\mu$  axis.

memory. From the second part of Lemma 4, if segmentation with parameter  $p$  is used to obtain a channel  $(b, g, \eta_B, \eta_G)$  with  $\mu = 1 - b - g$ , from a channel  $(b^*, g^*, \eta_B, \eta_G)$  with  $\mu^* = 1 - b^* - g^*$ , then it is straightforward to show that  $\mu = (1-p)\mu^*$ . Thus, from Theorem 6 and its corollary,  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$  is preserved for increasing  $\mu$ , while  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$  is preserved for decreasing  $\mu$ . Similarly to our arguments in the previous sections, by considering the line formed by fixing  $\eta_B$  and  $\eta_G$  and allowing  $\mu$  to vary, for a given point  $\mu$  on this line for which  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$ , all values of  $\mu^* > \mu$  will result in a one-dimensional decoding region. Combined with our earlier results for the plane of inversion probability pairs  $(\eta_B, \eta_G)$ , this result implies a three-dimensional decoding region in the GE parameter space, as the two-dimensional region is projected through decreasing  $\mu$ , depicted in Fig. 7. Conversely, for a given point  $\mu$  for which  $P_{\text{err}}[\ell_{\text{max}}] > \epsilon$ , a three-dimensional nondecoding region is obtained in conjunction with our earlier results.

#### IV. DENSITY EVOLUTION FOR GE-LDPC DECODERS

In Sections II and III, we established the feasibility and properties of a DE algorithm for GE-LDPC decoders. In this section, we obtain this algorithm, which is derived as a modification to the DE algorithm from [5]. We will first discuss the iterative construction of local neighborhoods, which will provide a basis for our derivation. Consider an edge  $e$  connecting variable  $U$  and factor  $v$ , and its local neighborhood  $\mathcal{N}_e^{2\ell}$ . Let  $\mathcal{L}_U$  be the set of edges attached to  $U$ , and let  $\mathcal{L}_U \setminus \{e\}$  be the set of all edges attached to  $U$  except  $e$ . Then assuming  $\mathcal{N}_e^{2\ell}$  is a tree, for each edge  $f$  in  $\mathcal{L}_U \setminus \{e\}$ , one may obtain the set of all paths of length  $2\ell - 1$  emanating from the factor attached to  $f$ , such that  $f$  is not the first edge traversed. If the edge  $f$  and the variable node  $u$  are appended to this set, we have a set which corresponds to

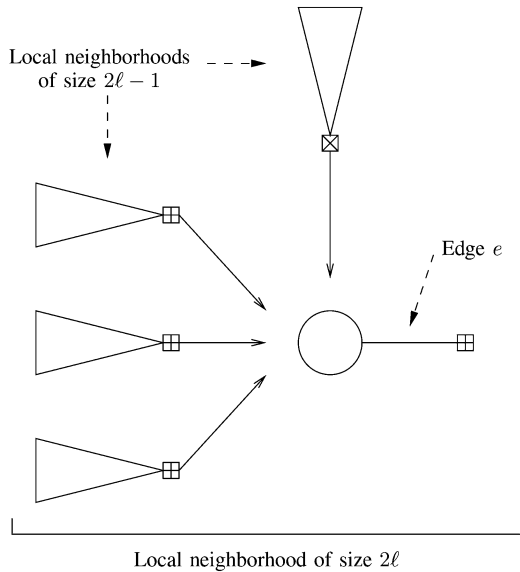


Fig. 8. Recursive construction of a local neighborhood  $\mathcal{N}_e^{2\ell}$  from local neighborhoods  $\mathcal{M}_f^{2\ell-1}$ . Notice that  $e$  connects a symbol variable and a parity check; the input local neighborhoods therefore include the output of a channel factor.

the definition of  $\mathcal{M}_f^{2\ell-1}$ . Furthermore, we have a (partial) set of paths of length  $2\ell$  emanating from  $U$ , such that  $e$  is not the first edge in the path.  $\mathcal{N}_e^{2\ell}$  is then obtained by taking the union of the set of neighborhoods

$$\bigcup_{f \in \mathcal{L}_U \setminus \{e\}} \mathcal{M}_f^{2\ell-1}$$

and appending the edge  $e$  and the factor  $v$ , noting that the variable  $U$  is common to all these local neighborhoods. A similar argument can be made in forming  $\mathcal{M}_e^{2\ell-1}$  from a union of  $\mathcal{N}_e^{2\ell-2}$ . Thus, local neighborhoods are constructed recursively, which is depicted in Fig. 8.

Since DE calculates the density of the message along the edge at the apex of the local neighborhood, this implies that DE is calculated iteratively, by taking the densities of all input messages to a node and applying the transformation implicit in the node from the SPA. We shall use this approach to obtain a DE algorithm for the GE-LDPC decoder. To decode the channel outputs provided by the model in (2), we use message-passing techniques, with the SPA to calculate messages. Because our objective is to calculate DE, our focus is on message passing within local neighborhoods, although our method can be straightforwardly applied to the more general task of decoding in the GE-LDPC factor graph. Furthermore, since we iteratively assemble local neighborhoods, we need only concern ourselves with the transformations at the common node in the assembly of new local neighborhoods. This allows a simplification of notation: we eliminate all time-index subscripts in message calculation; the assumed time index is the index of the common node, while all remaining subscripts correspond to the iteration number for DE.

The evolution of various kinds of messages in a local neighborhood must be tracked; these message types are differentiated by the types of nodes at each end of the apex edge of the local neighborhood, as we outline in what follows. Recall that

the message calculations were given in Section II. In the sequel, we will use an abbreviated notation for the message pdfs. For example, for the channel message  $C$ , the density is written  $f_C(c)$ , but we will write  $f_C$ . We will extend this notation by writing  $f_{C,j}$  to represent the density  $f_C(c)$  at the  $j$ th iteration of density evolution.

**Parity-check to symbol-variable messages (and vice versa).** The message-passing rules for messages exiting parity-check nodes and symbol-variable nodes are well known, and have been thoroughly discussed elsewhere (such as in [2], [3], [5]). Thus, density evolution is virtually the same as in the memoryless case. Let  $P$  and  $Q$  be, respectively, the messages at the output of a symbol-variable node and a parity-check node during decoding. From [5], in a memoryless channel, we have that  $f_{Q,j}$  is a complicated functional of  $f_{P,j}$ , and

$$f_{P,j+1} = \mathcal{F}^{-1}[\mathcal{F}(f_{P,0})(\mathcal{F}(f_{Q,j}))^{d_v-1}] \quad (11)$$

where  $\mathcal{F}$  represents the Fourier transform, and where  $d_v$  is the number of checks attached to a symbol variable node. Since the  $f_{P,0}$  term represents the channel information, in the case of a GE channel we replace this with the density of channel messages at each iteration, which is given below.

We briefly sketch the calculation of  $f_{Q,j}$  from  $f_{P,j}$ , as described in [5]. We know that the message  $Q$  is calculated from input messages  $P_i$  by

$$\tanh\left(\frac{Q}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{P_i}{2}\right).$$

The product becomes a sum by taking the log of both sides, and the density of the sum of the resulting random variables can be evaluated using the characteristic function, though this is complicated by the fact that both sides can take negative values. Define the ordered pair  $\tilde{P}_i := (\text{sign}(P_i), -\log \tanh(|P_i|/2))$ . Then using the generalized Fourier transform over  $\{-1, 1\} \times [0, \infty)$ , and using  $f(\tilde{p}_i)$  as the density of  $\tilde{P}_i$ , which can be obtained straightforwardly from  $f_P$ , we form an intermediate density  $\mathcal{F}^{-1}[\mathcal{F}(f(\tilde{P}_i))^{d_c-1}]$ . This density may then be converted to  $f_{Q,j}$  through a straightforward variable transformation.

**Symbol-variable to channel-factor messages.** This message, labeled  $D$ , conveys the code's extrinsic information to the Markov subgraph. As a log-likelihood ratio, it is calculated by taking the sum of all incoming parity-check to symbol-variable messages (taking care to exclude the channel message). The density of this message is designated  $f_{D,j}$  (for the  $j$ th iteration of density evolution), and the evolution of this message is calculated by

$$f_{D,j} = \mathcal{F}^{-1}[(\mathcal{F}(f_{Q,j}))^{d_v}] \quad (12)$$

in a manner similar to (11).

**Forward and backward messages.** The calculation of these messages, designated  $\mathbf{A}$  and  $\mathbf{B}$ , respectively, was given in (4) and (5). Since the vectors  $\mathbf{A}$  and  $\mathbf{B}$  are normalized as probabilities, once the value of one element is known, the other is fixed. Without loss of generality, they may be completely characterized as random variables by considering only the first element in each vector:  $A_1$  and  $B_1$ , respectively. We write their densities at the  $j$ th iteration as  $f_{A_1,j}$  and  $f_{B_1,j}$ , respectively.

Since  $A_1$  is a function of  $A_1^-$  and  $D$ , following our preceding discussion of assembling local neighborhoods, the density  $f_{A_1,j+1}$  is calculated from  $f_{A_1,j}$  and  $f_{D,j}$ . After some manipulation on (4), it is straightforward to show that the transformation from  $A_1^-$  to  $A_1$  is of the form

$$A_1 = \frac{N_1 + N_2 A_1^-}{D_1 + D_2 A_1^-} \quad (13)$$

where

$$\begin{aligned} N_1 &= g(\eta_B + (1 - 2\eta_B)\gamma(D, Y)) \\ N_2 &= (1 - b)(\eta_G + (1 - 2\eta_G)\gamma(D, Y)) \\ &\quad - g(\eta_B + (1 - 2\eta_B)\gamma(D, Y)) \\ D_1 &= \eta_B + (1 - 2\eta_B)\gamma(D, Y) \\ D_2 &= \eta_G + (1 - 2\eta_G)\gamma(D, Y) \\ &\quad - \eta_B - (1 - 2\eta_B)\gamma(D, Y) \end{aligned}$$

recalling the definition of  $\gamma$  from (3). Clearly,  $N_1, N_2, D_1,$  and  $D_2$  are merely functions of the random variables  $Y$  and  $D$ , which allows us to calculate a conditional probability transformation  $f(a_1|y, d)$  given these variables. However, notice that  $A_1^-$  and  $Y$  are dependent, since they are both related to the state  $S$  which gives rise to the noise element  $Z$ . This complication is removed if we obtain pdfs conditioned on the state  $S$  and subsequently marginalize, since  $A_1^-$  is a function of a set of observations  $\{Y_j\}$ , which is conditionally independent of  $Y$  given  $S$ . Using Bayes' rule, we first calculate

$$f(a_1^- | s) = \frac{p(s|a_1^-) f(a_1^-)}{p(s)} \quad (14)$$

where  $f(a_1^-) = f_{A_1,j}$ , and where  $p(s|a_1^-) = a_1^-$  if  $s = G$  and  $p(s|a_1^-) = 1 - a_1^-$  if  $s = B$ . With this quantity calculated, using a well-known result of functions of random variables (see, for example, [18, Ch. 4]), the transformation in (13) may be expressed as

$$f(a_1 | s, y, d) = f\left(\frac{D_1 A_1 - N_1}{N_2 - D_2 A_1} \middle| s\right) \cdot \left| \frac{D_1(N_2 - D_2 A_1) + D_2(D_1 A_1 - N_1)}{(N_2 - D_2 A_1)^2} \right| \quad (15)$$

which is marginalized with respect to  $S, Y,$  and  $D$ . Thanks to the conditioning on  $S$ , the marginalization is expressed simply by

$$f(a_1) = \sum_s \sum_y p(y|s)p(s) \int_d f(a_1 | s, y, d) f(d) dd$$

where  $f(d) = f_{D,j}$ , and where  $\Pr(y = 1|s) = \eta_s$  on the assumption that the all-zero codeword is transmitted. Finally,  $f_{A_1,j+1} = f(a_1)$ .

Similar expressions to (13)–(15) may be derived for the iteration from  $f_{B_1,j}$  and  $f_{D,j}$  to  $f_{B_1,j+1}$ . Again, the calculation of the message  $B_1$  is of the form

$$B_1 = \frac{N'_1 + N'_2 B_1^-}{D'_1 + D'_2 B_1^-} \quad (16)$$

where

$$\begin{aligned} N'_1 &= b(\eta_G + (1 - 2\eta_G)\gamma(D, Y)) \\ N'_2 &= (1 - 2b)(\eta_G + (1 - 2\eta_G)\gamma(D, Y)) \\ D'_1 &= b(\eta_G + (1 - 2\eta_G)\gamma(D, Y)) \\ &\quad + (1 - g)(\eta_B + (1 - 2\eta_B)\gamma(D, Y)) \\ D'_2 &= (1 - 2b)(\eta_G + (1 - 2\eta_G)\gamma(D, Y)) \\ &\quad - (1 - 2g)(\eta_B + (1 - 2\eta_B)\gamma(D, Y)). \end{aligned}$$

Finally, we must specify the calculation of messages from state-variable nodes to channel-factor nodes. Every state-variable node has degree 2, so by the SPA the input message is passed directly to the output without processing. No calculation or density evolution is required at these nodes.

**Channel messages.** This message, designated  $C$ , conveys the channel information from the Markov subgraph to the LDPC subgraph, and its density at the  $j$ th iteration of DE is given by  $f_{C,j}$ . The calculation of this message was given in (6).

The recursion of  $f_{C,j}$  from  $f_{A_1,j}$  and  $f_{B_1,j}$  can be obtained similarly to the above, with minor modifications on account of the logarithm. From (6), the message  $C$  is already a scalar, and can be expressed in the form

$$C = \sigma(Y) \log \frac{N_1^* + N_2^* A_1}{D_1^* + D_2^* A_1} \quad (17)$$

where  $\sigma(Y)$  is as defined in Section II, and

$$\begin{aligned} N_1^* &= (1 - \eta_B)(1 - g - B_1 + 2gB_1) \\ N_2^* &= (1 - \eta_G)(b + (1 - 2b)B_1) \\ &\quad - (1 - \eta_B)(1 - g - B_1 + 2gB_1) \\ D_1^* &= \eta_B(1 - g - B_1 + 2gB_1) \\ D_2^* &= \eta_G(b + (1 - 2b)B_1) - \eta_B(1 - g - B_1 + 2gB_1) \end{aligned}$$

so  $N_1^*, N_2^*, D_1^*,$  and  $D_2^*$  are functions of the random variable  $B_1$ , which allows us to specify a conditional density. Again, to simplify later calculations, we condition  $A_1$  and  $B_1$  on  $S$ , since these quantities are conditionally independent given  $S$ . We first obtain  $f(a_1|s)$ , as in (14), then apply the transformation required by (17), which results in

$$f(c | s, y, b_1) = f\left(\frac{D_1^* e^{\sigma(Y)C} - N_1^*}{N_2^* - D_2^* e^{\sigma(Y)C}} \middle| s\right) \cdot \left| \frac{D_1^*(N_2^* - D_2^* e^{\sigma(Y)C}) + D_2^*(D_1^* e^{\sigma(Y)C} - N_1^*)}{(N_2^* - D_2^* e^{\sigma(Y)C})^2} \right| e^{\sigma(Y)C} \quad (18)$$

and finally marginalize the conditional density, which results in

$$f(C) = \sum_s \sum_y p(y|s)p(s) \cdot \int_{b_1} f(c | s, y, b_1) f(b_1 | s) db_1$$

which gives  $f_{C,j}$ , the density that replaces  $f_{P,0}$  in (11).

The DE algorithm then proceeds as follows, in accordance with the message passing schedule defined in Section II.

1) **Initialization step.** Let  $j = 0$ . Set the following densities:

$$\begin{aligned} f_{A_1,0}(a_1) &= \delta(a_1 - g/(b + g)) \\ f_{B_1,0}(b_1) &= \delta(b_1 - 1/2) \\ f_{C,0}(c) &= \bar{\eta} \delta(c + \log(1 - \bar{\eta})/\bar{\eta}) \\ &\quad + (1 - \bar{\eta}) \delta(c - \log(1 - \bar{\eta})/\bar{\eta}) \\ f_{P,0} &= f_{C,0} \end{aligned}$$

where  $\delta(\cdot)$  is the Dirac delta function, and  $\bar{\eta}$  is the average inversion probability.

- 2) **Parity-check nodes.** Calculate  $f_{Q,j}$  from  $f_{P,j}$  according to the procedure outlined in [5].
- 3) **Channel-factor nodes.**
  - Calculate  $f_{A_{1,j+1}}$  from  $f_{A_{1,j}}$  and  $f_{D,j}$ , and calculate  $f_{B_{1,j+1}}$  from  $f_{B_{1,j}}$  and  $f_{D,j}$ , according to (15).
  - Calculate  $f_{C,j+1}$  from  $f_{A_{1,j}}$  and  $f_{B_{1,j}}$ , according to (18).
- 4) **Symbol-variable nodes.**
  - Similarly to (11), calculate

$$f_{P,j+1} = \mathcal{F}^{-1}[\mathcal{F}(f_{C,j})(\mathcal{F}(f_{S,j}))^{d_v-1}] \quad (19)$$

where  $d_v$  is the degree of the symbol-variable nodes, and the density  $f_{S,j}$  is calculated from  $f_{P,j}$  through the procedure outlined in [5].

- If  $j \geq 1$ , let

$$f_{D,j+1} = \mathcal{F}^{-1}[(\mathcal{F}(f_{Q,j}))^{d_v}] \quad (20)$$

else let  $f_{D,j+1} = f_{P,0}$ .

- 5) **Channel-state nodes.** No calculation required; messages from the channel factor nodes are unchanged.
- 6) Let  $j = j + 1$ . If  $j > \ell_{\max}$  or  $P_{\text{err}}[j] < \epsilon$ , stop; else go to 2).

As an aside, we briefly describe the modifications that are required to calculate DE for irregular LDPC codes; the reader is directed to [3] for more details. Let  $\lambda_i$  and  $\rho_i$  be the probabilities that a given edge in the LDPC subgraph is attached to a symbol-variable node of degree  $i$  and a parity-check node of degree  $i$ , respectively. Also, let  $\hat{\lambda}_i$  be the probability that a given symbol-variable node in the LDPC subgraph has degree  $i$ . Then (19) is replaced with

$$f_{P,j+1} = \mathcal{F}^{-1} \left[ \mathcal{F}(f_{C,j}) \sum_{i=1}^{v_{\max}} \lambda_i (\mathcal{F}(f_{Q,j}))^{i-1} \right]$$

where  $v_{\max}$  represents the maximum variable degree, and (20) is replaced with

$$f_{D,j+1} = \mathcal{F}^{-1} \left[ \sum_{i=1}^{v_{\max}} \hat{\lambda}_i (\mathcal{F}(f_{Q,j}))^i \right].$$

Meanwhile, let  $f_{Q,j}^{(i)}$  represent the density of a message at the output of a parity-check node of degree  $i$  after the  $j$ th iteration. These can be easily calculated from  $f_{P,j}$  using the procedure in [5], which we sketched earlier in this section. In step 2), we calculate  $f_{Q,j}$  using

$$f_{Q,j} = \sum_{i=2}^{c_{\max}} \rho_i f_{Q,j}^{(i-1)}$$

where  $c_{\max}$  is the maximum parity-check degree. However, we note that a concentration theorem (as in [3]) would be required to establish the accuracy of the DE results returned for irregular LDPC codes over the GE channel.

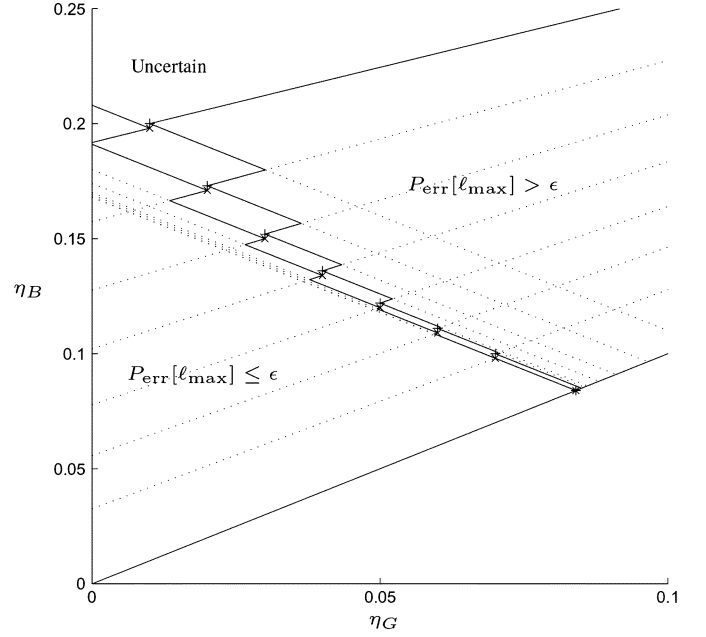


Fig. 9. Decoding and nondecoding regions for a GE-LDPC decoder with (3, 6) regular LDPC code. Points for which  $P_{\text{err}}[\ell_{\max}] \leq \epsilon$  are represented with “x,” and points for which  $P_{\text{err}}[\ell_{\max}] > \epsilon$  are represented with “+.” The dotted lines represent the individual decoding and nondecoding regions induced by each point, while the solid outline represents the union of these regions.

## V. DISCUSSION AND RESULTS

In this section, we consider the implementation of the DE algorithm introduced in Section IV, and the consequences of the theorems from Section III. We will first discuss the formation of a decoding region using the proposed DE algorithm. Suppose a set of channels representing points in  $\mathcal{G}$  are found by our DE algorithm to converge to  $P_{\text{err}}[\ell_{\max}] \leq \epsilon$ . Then the decoding region induced by these points is the *union* of the regions induced by each point. An example of such a union of regions is given in Fig. 9, for a (3, 6)-regular rate-1/2 LDPC code, where for ease of visualization we fix  $b = g = 0.01$ , so that  $\eta_B$  and  $\eta_G$  form a plane. From the figure, the space of parameters is partitioned into a region where  $P_{\text{err}} \rightarrow \epsilon$ , a region where  $P_{\text{err}} > \epsilon$ , and an uncertain region, which guides the selection of new points for DE. As  $\eta_G \rightarrow 0$ , we see from the figure that large gaps of uncertain decoding can form. This is because the “teeth” of the decoding regions are offset with respect to each other, and because they sweep out less area since they are closer to the boundary of  $\mathcal{G}$ . Thus, points at closer intervals should be evaluated as  $\eta_G \rightarrow 0$ .

A three-dimensional characterization of the decoding region is given in Fig. 10 for the (3, 6)-regular rate-1/2 LDPC code, and in Fig. 11 for the (3, 4)-regular rate-1/4 LDPC code. In these figures, we used  $\epsilon = 10^{-8}$  and  $\ell_{\max} = 1000$ . For clarity, we show only the decoding region, and trace the boundary on this region by joining points in the parameter space found to have  $P_{\text{err}} \rightarrow \epsilon$  (the assumed nondecoding region is the complement of this region). In both cases, we fix  $b/g = 1$  and depict the three-dimensional decoding region with respect to the triples  $(\mu, \eta_B, \eta_G)$ , where  $\mu = 1 - b - g$ . Tremendous possible gains in performance over the memoryless assumption are observed in Figs. 10 and 11. The performance improvement is particularly

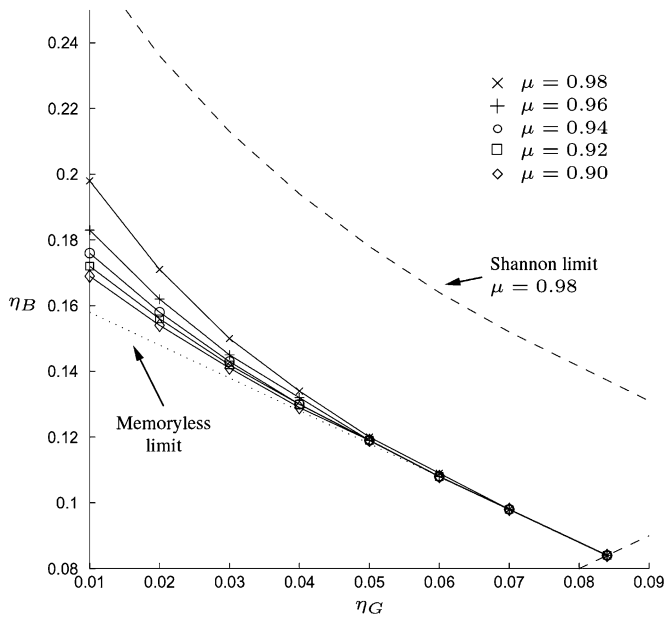


Fig. 10. Contour plot for (3, 6)-regular rate-1/2 code, looking down through the increasing  $\mu$  axis. The contours (solid lines) give the estimated boundary of the decoding region for constant  $\mu$ . The dashed line at top represents the Shannon limit for rate-1/2 codes at  $\mu = 0.98$ . The dotted line at bottom represents the limit as  $\mu \rightarrow 0$  (i.e., the memoryless limit).

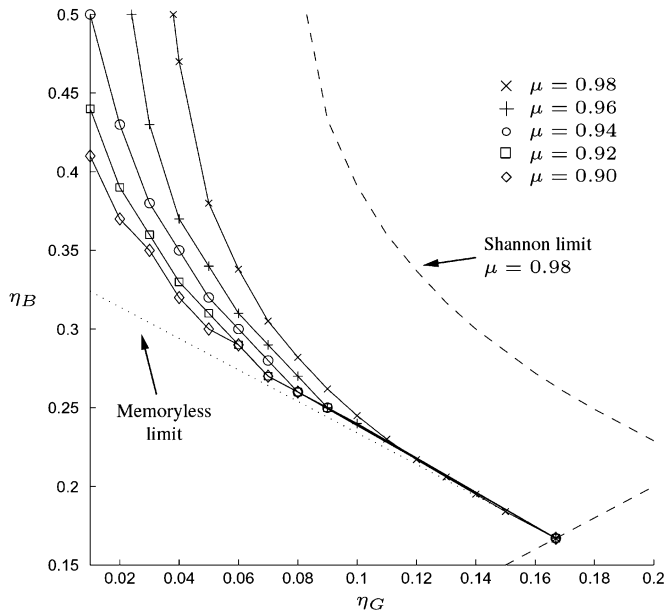


Fig. 11. Contour plot for (3, 4)-regular rate-1/4 code, looking down through the increasing  $\mu$  axis. The contours (solid lines) give the estimated boundary of the decoding region for constant  $\mu$ . The dashed line at top represents the Shannon limit for rate-1/4 codes at  $\mu = 0.98$ . The dotted line at bottom represents the limit as  $\mu \rightarrow 0$  (i.e., the memoryless limit).

noted when the contrast between  $\eta_B$  and  $\eta_G$  is large, and when  $\mu$  is very close to 1. This is understandable, since it is under these circumstances that the GE factor graph has the best ability to observe a state for a long period of time and differentiate between the two states.

Empirical results are presented in Fig. 12 which support our claims. Below the GE-LDPC decoding threshold, the decoder performance improves everywhere as block length increases. However, the effect of long channel memory is to increase the

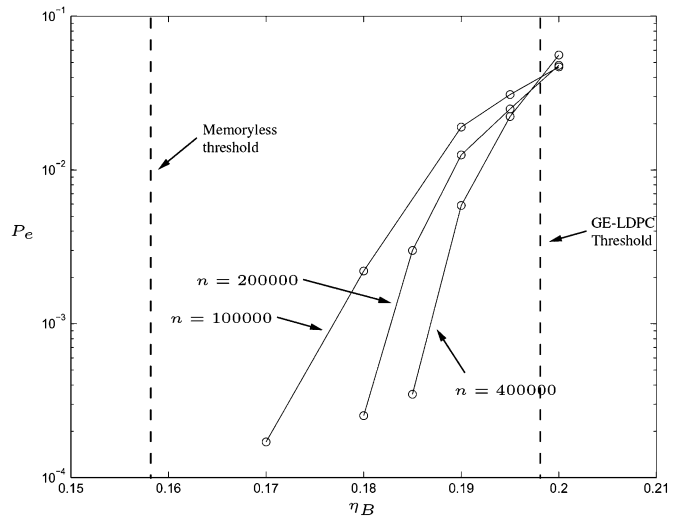


Fig. 12. Experimental results for a (3, 6)-regular rate-1/2 LDPC code, with  $b = 0.01, g = 0.01$ , and  $\eta_G = 0.01$ , for probability of symbol error versus  $\eta_B$ . Notice that the probability of error decreases everywhere below the GE-LDPC threshold as block length increases. For this family of channels, the Shannon limit for rate 1/2 is reached at  $\eta_B = 0.269$ , not shown on the figure.

block length required to approach the threshold performance, as observed in the figure. The assumption made in deriving the DE algorithm is that the block length is infinite, and although the GE-LDPC decoder clearly beats the memoryless threshold, it approaches the GE-LDPC threshold quite slowly—significantly more slowly than is the case in memoryless channels.

It is natural to compare these results to the channel capacity. In [19], the capacity of the GE channel was given as

$$C = \lim_{n \rightarrow \infty} E[1 - \mathcal{H}(\Pr(Z_n = 1 | \mathbf{Z}_{n-1} = \mathbf{z}_{n-1})))] \quad (21)$$

where  $\mathcal{H}(\cdot)$  is the binary entropy function, and  $\mathbf{Z}_{n-1}$  is the noise vector up to time  $n - 1$ . A significant gap to the GE channel capacity remains in each figure. However, in the case of the (3, 4) code, we observe that communication is possible with  $P_{\text{err}} \rightarrow \epsilon$  in channels where the  $C < 1/4$  for the memoryless assumption, which corresponds to an inversion probability of 0.214, an observation also made empirically for certain GE-LDPC decoders in [12]. Thus, the GE-LDPC decoder beats the best possible code which uses the memoryless assumption, which further justifies the use of joint estimation decoding in this channel. Whether capacity may be approached through the use of irregular LDPC codes remains an open problem.

It should be pointed out that our geometric results obtained in Section III are *sufficient* to show that a set of points is in the decoding region, but not *necessary*. In two dimensions, we have seen that each point for which  $P_{\text{err}}[\ell_{\text{max}}] \leq \epsilon$  induces a triangular decoding region, and from the figure the union of these regions produces a sawtooth-shaped region, and a similarly rough nondecoding region. In three dimensions, the union of these regions is no more smooth. However, we have observed through experimentation that the boundary on the decoding region may be traced with reasonable accuracy by simply joining adjacent points of successful decoding, as found by DE. We feel that a more accurate characterization of the parameter space, as well as any solution to the problem of expanding the existing decoding regions, may require proof techniques beyond the Side

Information Lemma. In fact, the Side Information Lemma may be a more stringent tool than necessary, since results following from this lemma imply that  $P_{\text{err}}[\ell]$  for one channel dominates  $P_{\text{err}}[\ell]$  for another channel at *every* value of  $\ell$ , not only in the limit of large  $\ell$ . Finding alternative strategies is a subject of ongoing research.

We finally note that our restriction to a three-dimensional induced decoding region may not represent an undue burden, since in many practical cases only the three parameters  $(\mu, \eta_B, \eta_G)$  are required to express a family of related channels. From [19], the space formed by pairs of state transition probabilities  $(b, g)$  may be expressed as pairs  $(\rho, \mu)$ , where  $\rho := g/b$ , representing the overall ratio of state  $G$  to state  $B$  in the state sequence. By varying  $\mu$  and fixing  $\rho$ , we may represent the practically interesting case of increasing the symbol rate in a physical system. For example, if the GE channel is the model for a system in which the symbol transmission rate can be varied, and the channel state process can be modeled as an independent process sampled at the transmission rate, we would expect the proportion of  $B$  and  $G$  states to remain the same (i.e., constant  $\rho$ ), while the observed channel memory would increase as a result of the increased symbol rate (increasing  $\mu$ ), which from Theorem 6 is known to improve the probability of error.

In forthcoming work, we intend to build on the foundation laid in this paper, extending these results by generalizing the decoding region theorems to discrete-valued Markov channels with an arbitrary number of states, in addition to continuous-valued Markov channels, which are of much practical interest. To accomplish these tasks, analysis could be done similar to that in this paper, though with the complication that the channel messages would be passed as vectors, and thus would have multivariate densities. Alternatively, a Monte Carlo method could be used to obtain the channel message densities without directly calculating DE through the Markov chain. Work is also progressing in extending these ideas to the optimization of irregular LDPC codes, which have been shown in memoryless channels to have capacity-approaching performance. Clearly, the work in this paper has demonstrated that the strategy of joint estimation decoding for LDPC codes in the GE channel is *quantifiably* superior to ignoring or destroying the channel memory. This was done by providing the analytical tools to obtain the ultimate performance of such decoders, and by applying these tools in a variety of instances of the GE channel. As well, we have demonstrated that DE is a feasible analytical tool in the GE channel, since the complexity of multidimensional parameter spaces has been successfully mitigated through theoretical analysis of decoding regions.

APPENDIX

A. Proof of Theorem 5

In this part of the appendix, we complete the Proof of Theorem 5 by showing that channel outputs in a local neighborhood in the factor graph for a GE channel with inversion probabilities  $(\eta_B, \eta_G)$ , combined with carefully chosen side information, result in a factor graph which is identical to a local neighborhood with inversion probabilities  $(\eta_B^*, \eta_G^*)$ , where the

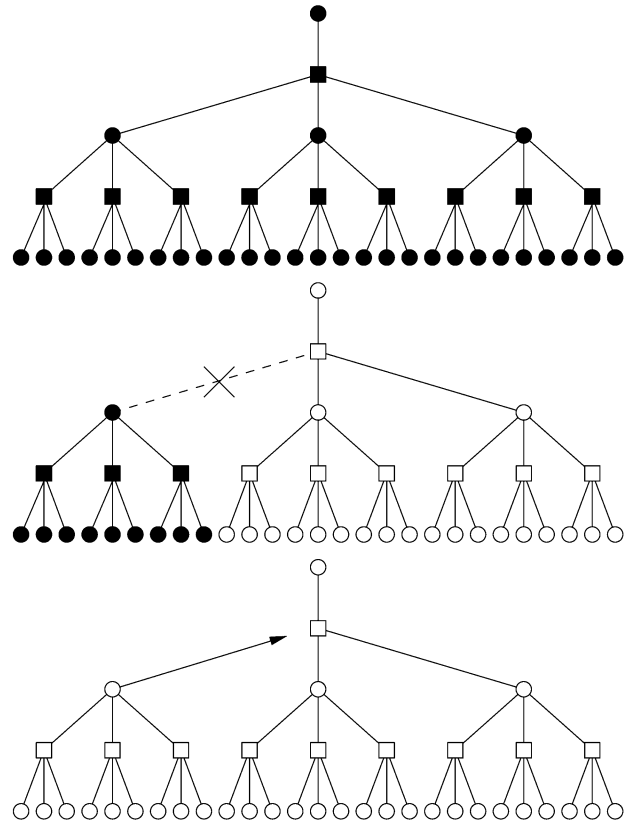


Fig. 13. Illustration of the proof technique. Consider two channels  $c^*$  and  $c$ , where side information may be used to transform  $c$  into  $c^*$ . Start with a local neighborhood for  $c$  (top). Partial side information disconnects the factor graph, and the nodes left behind are a subgraph of a local neighborhood for  $c^*$ , represented by white nodes (middle). Using the missing nodes as a generative model, additional side information completes the local neighborhood (bottom).

pair  $(\eta_B, \eta_G)$  arose from a channel with inversion probabilities  $(\eta_B^*, \eta_G^*)$  applied to a state scrambler with parameter  $\psi$ , and where the scrambling sequence is  $\mathbf{R}$ . Under the independence assumption (which implies that the local neighborhood is cycle-free), the local neighborhood implements an optimal detector for the symbol at the apex of the local neighborhood, so the side information lemma may be applied. The general proof technique is illustrated in Fig. 13.

The state transition probabilities  $(b, g)$  are fixed. Let  $\mathcal{N}_e^{2\ell}$  represent the cycle-free local neighborhood to depth  $2\ell$  of an edge  $e$  from a GE-LDPC factor graph with channel inversion probability pair  $(\eta_B, \eta_G)$ , and let  $\mathbf{Y}_e, \mathbf{X}_e$ , and  $\mathbf{S}_e$  represent the observations, symbol variables, and channel states associated with this local neighborhood, respectively. Then the pmf of  $\mathbf{Y}_e, \mathbf{X}_e$ , and  $\mathbf{S}_e$  may be written as

$$p(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) = u \prod_{i \in \mathcal{I}} p(s_{i+1} | s_i) \cdot \prod_{j \in \mathcal{J}} p(y_j | x_j, s_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (22)$$

where  $\mathcal{I}, \mathcal{J}$ , and  $\mathcal{K}$  are, respectively, index sets for the factors  $p(s_{i+1} | s_i), p(y_j | x_j, s_j)$ , and  $h_k(\mathbf{x}_k)$  in  $\mathcal{N}_e^{2\ell}$ ; and  $u$  is a constant. The variable node  $\hat{X}$  connected to edge  $e$  is the objective for the local neighborhood, so the decoding task is to calculate the *a posteriori* probability of  $\hat{X}$ , using the SPA. We wish to

show that side information transforms the model in (22) to a model equivalent to

$$p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) = u^* \prod_{i \in \mathcal{I}} p(s_{i+1} | s_i) \cdot \prod_{j \in \mathcal{J}} p^*(y_j | x_j, s_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (23)$$

where  $p^*(y_j | x_j, s_j)$  implements the GE inversion probabilities  $(\eta_G^*, \eta_B^*)$ .

We provide side information to the model in (22) in two steps. First, suppose the scrambling sequence  $\mathbf{R} = \mathbf{r}$  is given, and let  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e, \mathbf{s}'_e)$  be the local neighborhood pmf with this information, noting that this local neighborhood now depends on the scrambled states  $\mathbf{S}'_e$ . Then (22) becomes

$$p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e, \mathbf{s}'_e) = u \prod_{i \in \mathcal{I}} p(s_{i+1} | s_i) \cdot \prod_{j \in \mathcal{J}} p(y_j | r_j, x_j, s_j, s'_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (24)$$

where  $p(y_j | r_j, x_j, s_j, s'_j)$  is defined as follows:

$$\begin{aligned} p(y_j | r_j = 0, x_j, s_j, s'_j) &= p^*(y_j | x_j, s_j) \\ &= \begin{cases} \eta_{s_j}, & x_j \neq y_j \\ 1 - \eta_{s_j}, & x_j = y_j \end{cases} \\ p(y_j | r_j = 1, x_j, s_j, s'_j) &= p^*(y_j | x_j, s'_j) \\ &= \begin{cases} \eta_{s'_j}, & x_j \neq y_j \\ 1 - \eta_{s'_j}, & x_j = y_j \end{cases} \end{aligned}$$

where, as before,  $p^*(y_j | x_j, s_j)$  is the conditional inversion probability of the  $(\eta_B^*, \eta_G^*)$  GE channel. In other words, when  $r_j = 0$ , the original inversion probabilities  $(\eta_B^*, \eta_G^*)$  are a function of  $S_j$ , and when  $r_j = 1$ , these probabilities are a function  $S'_j$ . Furthermore, if  $r_j = 1$ , the term  $p(y_j | r_j = 1, x_j, s_j, s'_j)$  is independent of  $S_j$ . Since the channel factor node function  $p(s_{j+1} | s_j) p^*(y_j | x_j, s'_j)$  then factors into independent terms, this implies that an edge in the local neighborhood is broken for each  $r_j = 1$ . As well, since the factor graph of  $\mathcal{N}_{\bar{e}}^{2\ell}$  is cycle free, this implies the resulting factor graph is *disconnected*; that is, the model can be factored into terms that represent independent pmfs. Since these terms share no variables, their factor graphs share no edges. Thus, there exist index subsets  $\bar{\mathcal{I}}_{\mathbf{r}} \subset \mathcal{I}$ ,  $\bar{\mathcal{J}}_{\mathbf{r}} \subset \mathcal{J}$ , and  $\bar{\mathcal{K}}_{\mathbf{r}} \subset \mathcal{K}$ , each dependent on  $\mathbf{r}$ , such that no factor indexed by any of  $\bar{\mathcal{I}}_{\mathbf{r}}$ ,  $\bar{\mathcal{J}}_{\mathbf{r}}$ , or  $\bar{\mathcal{K}}_{\mathbf{r}}$  contains  $\hat{X}$ , nor does a path exist from any of them to  $\hat{X}$  in the factor graph; and

$$\begin{aligned} p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e, \mathbf{s}'_e) &= C \left( \prod_{i \in \bar{\mathcal{I}}_{\mathbf{r}}} p(s_{i+1} | s_i) \right. \\ &\cdot \left. \prod_{j \in \bar{\mathcal{J}}_{\mathbf{r}}} p(y_j | x_j, s_j, s'_j, r_j) \prod_{k \in \bar{\mathcal{K}}_{\mathbf{r}}} h_k(\mathbf{x}_k) \right) \\ &\cdot \prod_{i \in \{\mathcal{I} \setminus \bar{\mathcal{I}}_{\mathbf{r}}\}} p(s_{i+1} | s_i) \prod_{j \in \{\mathcal{J} \setminus \bar{\mathcal{J}}_{\mathbf{r}}\}} p(y_j | x_j, s_j, s'_j, r_j) \\ &\cdot \prod_{k \in \{\mathcal{K} \setminus \bar{\mathcal{K}}_{\mathbf{r}}\}} h_k(\mathbf{x}_k) \end{aligned}$$

where the bracketed term is an independent factor. Since this factor does not contain  $\hat{X}$ , it is eliminated by the SPA with no effect on messages passed to  $\hat{X}$ , and may be disregarded.

We can now rewrite every term  $p(y_j | r_j, x_j, s_j, s'_j)$  as a function of either  $S_j$  alone or  $S'_j$  alone. Furthermore, the set  $\bar{\mathcal{J}}_{\mathbf{r}}$  normally contains indices corresponding to both  $S_j$  and  $S'_j$ . Since we eventually wish to show that this model is equivalent to (23), we will relabel all  $S'_j, j \in \mathcal{J} \setminus \bar{\mathcal{J}}_{\mathbf{r}}$  to  $S_j$ , noting that this relabeling does not affect the structure of the model. Thus, we have

$$\begin{aligned} p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) &= v \prod_{i \in \{\mathcal{I} \setminus \bar{\mathcal{I}}_{\mathbf{r}}\}} p(s_{i+1} | s_i) \\ &\cdot \prod_{j \in \{\mathcal{J} \setminus \bar{\mathcal{J}}_{\mathbf{r}}\}} p^*(y_j | x_j, s_j) \\ &\cdot \prod_{k \in \{\mathcal{K} \setminus \bar{\mathcal{K}}_{\mathbf{r}}\}} h_k(\mathbf{x}_k) \end{aligned}$$

and since  $p^*(y_j | x_j, s_j)$  implements the GE inversion probability pair  $(\eta_B^*, \eta_G^*)$ , the model represented by  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  contains a *subset* of the factors in the model  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  from (23). To complete the proof, we add further side information, as follows. Consider the set of factors in  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  that are not present in the above representation of  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$ . Generate side information with a pmf precisely equal to the observations in the “missing” factors from  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  (that is, these missing factors are a *generative model*). More precisely, let  $Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{r})$  be a generative model, such that

$$\begin{aligned} Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{r}) &= v^* \prod_{i \in \bar{\mathcal{I}}_{\mathbf{r}}} p(\bar{s}_{i+1} | \bar{s}_i) \prod_{j \in \bar{\mathcal{J}}_{\mathbf{r}}} p^*(\bar{y}_j | \bar{x}_j, \bar{s}_j) \prod_{k \in \bar{\mathcal{K}}_{\mathbf{r}}} h_k(\bar{\mathbf{x}}_k) \end{aligned}$$

where  $\bar{\mathbf{Y}}$ ,  $\bar{\mathbf{X}}$ , and  $\bar{\mathbf{S}}$  are the observations, symbols, and channel states in the generative model. Thus, we have chosen side information whose factor-graph model  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{r})$  is equivalent to  $M^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$ . Then the set of observations  $\{\mathbf{Y}_e, \mathbf{R}, \bar{\mathbf{Y}}\}$  results in a decoder equivalent to  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$ , and applying the SPA clearly results in identical results in either  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  or  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{r})$ . Furthermore, for each  $\mathbf{r}$ ,  $p^*(\mathbf{r}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{r})$  is identical to  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$ , so this is also true for every  $\mathbf{r}$ . By the Side Information Lemma this is sufficient to show that

$$P_{\text{err}}[\ell](b, g, \eta_B^*, \eta_G^*) \leq P_{\text{err}}[\ell](b, g, \eta_B, \eta_G)$$

which completes the proof.

### B. Proof of Theorem 6

In this part of the appendix, we complete the Proof of Theorem 6. Since the proof proceeds almost identically to the Proof of Theorem 5, we will merely sketch this proof. Recalling the definitions for  $\Xi$ ,  $\mathbf{S}^*$ , and  $\mathbf{U}$ , let  $\mathcal{N}_{\bar{e}}^{2\ell}$  represent the cycle-free local neighborhood to depth  $2\ell$  of an edge  $e$  from a GE-LDPC factor graph with state transition probabilities governed by  $(b, g)$ , and again let  $\mathbf{Y}_e$  represent the observations associated with this local neighborhood. Then, the probabilistic model for  $\mathbf{Y}_e, \mathbf{X}_e$ , and  $\mathbf{S}_e$  may be written similarly as

$$\begin{aligned} p(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) &= u \prod_{i \in \mathcal{I}} p(s_{i+1} | s_i) \prod_{j \in \mathcal{J}} p(y_j | x_j, s_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (25) \end{aligned}$$

while the objective is again to obtain

$$p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) = u^* \prod_{i \in \mathcal{I}} p^*(s_{i+1} | s_i) \prod_{j \in \mathcal{J}} p(y_j | x_j, s_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (26)$$

where  $p^*(s_{i+1} | s_i)$  is the state transition probability governed by  $(b^*, g^*)$ . Providing the segmentation sequence  $\mathbf{U}$  to the decoder, the model becomes

$$p^*(\mathbf{u}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e) = u^* \prod_{i \in \mathcal{I}} p(s_{i+1} | s_i, u_i) \prod_{j \in \mathcal{J}} p(y_j | x_j, s_j) \prod_{k \in \mathcal{K}} h_k(\mathbf{x}_k) \quad (27)$$

where

$$\begin{aligned} p(s_{i+1} | s_i, u_i = 0) &= p^*(s_{i+1} | s_i) \\ &= \begin{cases} 1 - g, & s_i = B, & s_{i+1} = B \\ g, & s_i = B, & s_{i+1} = G \\ b, & s_i = G, & s_{i+1} = B \\ 1 - b, & s_i = G, & s_{i+1} = G \end{cases} \\ p(s_{i+1} | s_i, u_i = 1) &= p(s_{i+1}) \\ &= \begin{cases} b/(b+g), & s_{i+1} = B \\ g/(b+g), & s_{i+1} = G \end{cases} \end{aligned}$$

and where the former term is the same as the unsegmented channel, and the latter is independent of  $S_i$ . Again we have a break in the factor graph, disconnecting nodes wherever  $U_i = 1$ . The remainder of the argument follows identically to the Proof for Theorem 5: the model  $p^*(\mathbf{u}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  factors into independent terms, all but one of which may be discarded, while replacing the factors with further side information with model  $Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{u})$ , constructed from the “missing” factors in  $p^*(\mathbf{u}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$  results in a model  $p^*(\mathbf{u}, \mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)Z(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{s}}, \mathbf{u})$  which is equivalent to  $p^*(\mathbf{y}_e, \mathbf{x}_e, \mathbf{s}_e)$ . By the Side Information Lemma, this is sufficient to show that

$$P_{\text{err}}[\ell](b^*, g^*, \eta_B, \eta_G) \leq P_{\text{err}}[\ell](b, g, \eta_B, \eta_G).$$

#### ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their many helpful comments. They would also like to thank Chris Nicola of Queen’s University for his helpful comments on an early version of this paper.

#### REFERENCES

- [1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.
- [2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [3] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [4] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. L. 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.
- [5] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [6] J. Hou, P. H. Siegel, and L. B. Milstein, “Performance analysis and code optimization of low-density parity-check codes on rayleigh fading channels,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 924–934, May 2001.
- [7] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.
- [8] A. P. Worthen and W. E. Stark, “Low-density parity check codes for fading channels with memory,” in *Proc. 36th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sep. 1998, pp. 117–125.
- [9] A. P. Worthen and W. E. Stark, “Unified design of iterative receivers using factor graphs,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.
- [10] S. German and K. Kochanek, “Dynamic programming and the graphical representation of error-correcting codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 549–568, Feb. 2001.
- [11] J. Garcia-Frias and J. D. Villasenor, “Turbo decoding of Gilbert-Elliott channels,” *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 357–363, Mar. 2002.
- [12] J. Garcia-Frias, “Decoding of low-density parity-check codes over finite-state binary Markov channels,” *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1840–1843, Nov. 2004.
- [13] T. Wadayama, “An iterative decoding algorithm of low density parity check codes for hidden Markov noise channels,” in *Proc. IEEE Int. Symp. Information Theory and Its Applications*, Honolulu, HI, Nov. 2000.
- [14] A. Kavčić, X. Ma, and M. Mitzenmacher, “Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds,” *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1636–1652, Jul. 2003.
- [15] J. Li, K. R. Narayanan, E. Kurtas, and C. N. Georghiades, “On the performance of high-rate TPC/SPC codes and LDPC codes over partial response channels,” *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 723–734, May 2002.
- [16] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
- [17] N. Varnica and A. Kavčić, “Optimized LDPC codes for partial response channels,” in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, Jun. 2000, p. 197.
- [18] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. Menlo Park, CA: Addison-Wesley, 1994.
- [19] M. Mushkin and I. Bar-David, “Capacity and coding for the Gilbert-Elliott channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277–90, Nov. 1989.