Analysis and Design of Binary Message-Passing Decoders

Gottfried Lechner, Troels Pedersen, and Gerhard Kramer

Abstract—Binary message-passing decoders for low-density parity-check (LDPC) codes are studied by using extrinsic information transfer (EXIT) charts. The channel delivers hard or soft decisions and the variable node decoder performs all computations in the L-value domain. A hard decision channel results in the well-know Gallager B algorithm, and increasing the output alphabet from hard decisions to two bits yields a gain of more than 1.0 dB in the required signal to noise ratio when using optimized codes. The code optimization requires adapting the mixing property of EXIT functions to the case of binary message-passing decoders. Finally, it is shown that errors on cycles consisting only of degree two and three variable nodes cannot be corrected and a necessary and sufficient condition for the existence of a cycle-free subgraph is derived.

Index Terms—extrinsic information transfer charts, Gallager B Algorithm, irregular codes, low-density parity-check codes, message-passing decoding

I. INTRODUCTION

Gallager introduced low-density parity-check (LDPC) codes [1], [2] and also presented message-passing decoding algorithms that exchange only binary messages between the variable and check nodes. These algorithms are referred to as Gallager A and Gallager B [3] depending on how the variable-to-check node messages are computed. The algorithms have small memory requirements and low complexity implementations, especially of the check node decoder, and they have found practical use in high-speed applications, e.g. optical transmission systems [4]. However, the complexity advantages come at the cost of a significant loss in performance.

In this work, we use extrinsic information transfer (EXIT) charts [1], [2], [3], [4], [5] to analyze and design binary message-passing algorithms. Interestingly, in contrast to non-binary message passing where they are an approximation, EXIT charts are exact for the case of binary messages (and infinite-length code ensembles) since the mutual information describes the probability densities of the messages precisely. Furthermore, the EXIT functions for binary message-passing algorithms can be derived analytically [1], [2].

Binary message-passing algorithms were studied in [7], [8] where the authors showed that optimum algorithms must satisfy certain symmetry and isotropy conditions. In contrast to majority based decision rules, we assume that the variable node decoder converts all incoming messages to L-values [9], performs decoding in the L-value domain and applies a hard decision on the result. Note that for these algorithms, there always exist majority decision rules that can be derived in a straightforward way as shown in Section V-B. This general approach assures that the symmetry and isotropy conditions are satisfied and the algorithms can be extended for systems where the channel provides more information than hard decisions, while the variable and check node decoder still exchange binary messages only. This reduces the gap between optimum decoding and binary message-passing decoding, while the complexity is kept low.

Our main contributions are as follows:

- We derive a framework that allows binary message-passing algorithms to incorporate various quantization schemes for the channel messages. Increasing the number of quantization bits from one to two leads to a significantly improved decoding performance.
- We identify certain structures of the factor graph which cannot be corrected by binary message-passing decoders. Therefore, in addition to the stability condition, the degree distribution has to satisfy another constraint in order to avoid error floors. An important consequence is that regular LDPC codes with variable node degree three cannot be decoded without an error floor.

The rest of this paper is organized as follows. In Section II we introduce basics and definitions which are used in Section III to derive the EXIT functions of the variable and check node decoders. In Section IV, we show how the EXIT functions can be used to optimize the code and derive constraints on the degree distributions. Section V considers practical aspects of binary message-passing decoders and simulation results are presented in Section VI.

II. PRELIMINARIES

For binary message-passing decoders, the extrinsic channel [6] of the variable and check node decoder is represented as a binary symmetric channel (BSC) with input $X$ and output $Y$ both with alphabet $\{+1, -1\}$. Let $\epsilon$ denote the crossover probability of the BSC which we assume to be less than or equal to 0.5. Since there is a one-to-one relation between mutual information $I(X; Y)$ and crossover probability for the BSC, we can equivalently describe those channels using

$$I(X; Y) = 1 - h_0(\epsilon)$$  \hspace{1cm} (1)
where \( h_b(\cdot) \) denotes the binary entropy function
\[
h_b(\epsilon) = -\epsilon \log_2(\epsilon) - (1 - \epsilon) \log_2(1 - \epsilon). \tag{2}
\]
The variable node decoder converts all messages to L-values using
\[
L(y) = \log \frac{\Pr[X = -1|Y = y]}{\Pr[X = 1|Y = y]} = \log \frac{\Pr[Y = y|X = +1]}{\Pr[Y = y|X = -1]}, \tag{3}
\]
where \( y \) is a realization of \( Y \) and we assumed that \( \Pr[X = +1] = \Pr[X = -1] = 1/2 \). Defining the reliability associated with a BSC as
\[
D = \log \frac{1 - \epsilon}{\epsilon} \geq 0 \tag{4}
\]
allows to express the L-value as
\[
L(y) = y \cdot D. \tag{5}
\]
Throughout the paper, random variables are denoted by upper-case letters and their realizations are denoted by lowercase letters. The indices \( v, c, a \), and \( e \) stand for variable node decoder, check node decoder, a-priori and extrinsic, respectively.

### III. EXIT FUNCTIONS OF COMPONENT DECODERS

#### A. Check Node Decoder

A check node of degree \( d_c \) of a binary message-passing algorithm computes the output message of each of its edges as the product of the other \( d_c - 1 \) edge inputs when using the alphabet \{1, -1\}. Let \( \epsilon_{ac} \) denote the average bit error probability at the input of the check nodes. The corresponding a-priori crossover probability of the extrinsic channel is therefore also \( \epsilon_{ac} \). We define \( I_{ac} = 1 - h_b(\epsilon_{ac}) \) so that \( \epsilon_{ac} = h_b^{-1}(1 - I_{ac}) \) where \( h_b^{-1}(x) \) takes on values in the interval \([0, 1/2]\). The crossover probability at the check node output is \([2, \text{Lemma 4.1}]\)
\[
\epsilon_{ec} = f_c(\epsilon_{ac}; d_c) = \frac{1 - (1 - 2\epsilon_{ac})d_c - 1}{2} \tag{6}
\]
where \( f_c \) is the EXIT function of a check node of degree \( d_c \). Using \([6] \text{ and } [1]\), we define \( I_{ac} = 1 - h_b(\epsilon_{ac}) \). The inverse of the EXIT function in \((6)\) reads
\[
\epsilon_{ac} = f_c^{-1}(\epsilon_{ec}; d_c) = \frac{1 - (1 - 2\epsilon_{ec}) \frac{1}{2}}{2}. \tag{7}
\]

#### B. Variable Node Decoder

For a variable node of degree \( d_v \), every outgoing message \( L_{ev,j} \) along edge \( j \) of the variable node is given by
\[
L_{ev,j} = L_{ch} + \sum_{i=1,i\neq j}^{d_v} L_{av,i}, \quad j = 1, \ldots, d_v \tag{8}
\]
where \( L_{ch} \) is the L-value from the channel (see \([2]\) below) and \( L_{av,i} \) is the L-value from the check nodes along edge \( i \) of the variable node. To perform this summation, all messages must be converted to L-values using \([4] \text{ and } [5]\), where we assume that the variable node decoder knows the parameters of the communication and extrinsic channels. We show in Section \([VI]\) how the decoder can be implemented without this knowledge. In the following, we assume the communications channel is an additive white Gaussian noise channel with binary input (BIAWGN) and noise variance \( \sigma_n^2 \) so the received values \( y \) are converted to L-values as
\[
L_{ch} = \frac{2}{\sigma_n^2} y \tag{9}
\]
before being quantized. Conditioned on \( X = +1 \), the unquantized L-values are Gaussian random variables with variance \( \sigma_{ch}^2 = 4/\sigma_n^2 \) and mean \( \mu_{ch} = \sigma_{ch}^2/2 \tag{10} \). In the following, we derive the EXIT functions for three different quantization schemes.

#### 1) Hard Decision Channel: Consider the case where the receiver performs hard decisions. Then the decoder’s communication channel can be modeled as a BSC with crossover probability
\[
\epsilon_{ch} = \int_{-\infty}^{0} g(l) dl \tag{10}
\]
where \( g(l) = p_t (1 + l) \) is the conditional transition probability of the actual communication channel. Any symmetric channel with binary input is completely described by \( g(l) \) which is therefore sufficient to analyze the decoding behavior. Let \( D_{ch} \) and \( D_{av} \) denote the reliabilities of the decoder’s communication and extrinsic channels, respectively (see \([6, \text{Fig. 2 and Fig. 3}]\)). The variable node decoder computes the outgoing message on edge \( j \) by using \([8]\) with \( L_{ch} = y \cdot D_{ch} \). The outgoing message transmitted to the check node decoder is the sign of \( L_{ev,j} \). To compute the error probability of the message, we consider two cases. First, assume that the channel message is in error. This error is corrected if the sum over the \( L_{av,i} \) in \([8]\) can overcome the incorrect sign on \( L_{ch} \), i.e., if
\[
-D_{ch} - n_c D_{av} + (d_v - 1 - n_c) D_{av} \geq 0 \tag{11}
\]
where \( n_c \) is the number of erroneous messages from the check nodes. An equivalent condition is \( n_c \leq t \) where
\[
t = \left[ \frac{D_{av}(d_v - 1) - D_{ch}}{2D_{av}} \right]. \tag{12}
\]
Similarly, if the channel message is correct, then \( n_c \) has to be less than or equal to
\[
\bar{t} = \left[ \frac{D_{av}(d_v - 1) + D_{ch}}{2D_{av}} \right] \tag{13}
\]
result to be in a correct outgoing message. Combining these two cases yields the error probability of the outgoing messages of the variable node decoder
\[
\epsilon_{ev} = f_v(\epsilon_{av}; d_v, \epsilon_{ch}) = 1 - \epsilon_{ch} B(t; d_v - 1, \epsilon_{av}) - (1 - \epsilon_{ch}) B(\bar{t}; d_v - 1, \epsilon_{av}) \tag{14}
\]
where
\[
B(k; n, p) = \sum_{i=0}^{k} \binom{n}{i} p^i (1 - p)^{n-i} \tag{15}
\]
denotes the binomial cumulative distribution. The EXIT function (14) serves as a lower bound on EXIT functions for (appropriately designed) soft-decision detectors, see Section III-C below.

2) Soft Decision Channel: In the limit of no quantization of the output of a BIAWGN channel, the crossover probability at the output of the variable node is

\[ \epsilon_{ev} = Pr[L_{ev,j} \leq 0 | X = +1] \]

\[ = Pr\left[ L_{ch} + \sum_{i=1, i \neq j}^{d_v} L_{av,i} \leq 0 | X = +1 \right] \]

\[ = \sum_{z=0}^{d_v-1} Pr[N_c = z], \]

\[ Pr\left[ -L_{ch} + \mu_{ch} \geq D_{av}(d_v - 1 - 2z) + \mu_{ch} | X = +1 \right] \]

\[ = \sum_{z=0}^{d_v-1} b(z; d_v - 1, \epsilon_{av}) Q \left( \frac{D_{av}(d_v - 1 - 2z) + \mu_{ch}}{\sigma_{ch}} \right), \]

(16)

where \( N_c \) is a random variable representing the number of erroneous messages from the check nodes, where

\[ Q(\phi) = \frac{1}{\sqrt{2\pi}} \int_{\phi}^{\infty} e^{-x^2/2} dx \]

is the familiar \( Q \) function, and where

\[ b(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k} \]

(18)

denotes the binomial probability mass function. The EXIT function (16) serves as an upper bound on the EXIT function of any quantization scheme, see Section III-C below.

3) Output Alphabets Larger than Binary: Suppose a quantizer provides the sign \( sign(L_{ch}) \) and a quantization index \( w, w = 0, \ldots, W \), of the magnitude \( |L_{ch}| \). The boundary points of the quantizer are defined by the vector \( \zeta = [\zeta_0, \ldots, \zeta_W] \) where \( 0 < \zeta_0 < \zeta_1 < \cdots < \zeta_W \). Such a quantization scheme is depicted in Figure 1.

Following [11], [12], this channel quantization scheme can be decomposed as \((W + 1) \) BSCs. Sub-channel \( w \) is used with probability \( p_w \), has cross-over probability \( \epsilon_{ch,w} \) and reliability

\[ D_{ch,w}. \]

For \( w > 0 \) we have

\[ p_w = \begin{cases} \zeta_w & \text{if } 0 \leq \zeta_w \leq 1 \\ \zeta_{w-1} & \text{if } -1 \leq \zeta_{w-1} \leq 0 \end{cases} g(l)dl, \]

(19)

\[ \epsilon_{ch,w} = \begin{cases} 1 & \text{if } -1 \leq \zeta_{w-1} \leq 0 \\ \frac{1}{2} & \text{if } 0 < \zeta_w \leq 1 \end{cases}, \]

(20)

\[ D_{ch,w} = \log \frac{1 - \epsilon_{ch,w}}{\epsilon_{ch,w}}. \]

(21)

We define sub-channel zero \(( w = 0) \) as a BSC with crossover probability 0.5 [11], [12]. The parameters for sub-channel zero are

\[ p_0 = \int_{\zeta_0}^0 g(l)dl, \]

(22)

\[ \epsilon_{ch,0} = \frac{1}{2}, \]

(23)

\[ D_{ch,0} = 0. \]

(24)

The EXIT function of the overall channel is given by the expectation of the EXIT functions of the sub-channels [11], [12]:

\[ \epsilon_{ev} = \sum_{w=0}^{W} p_w \epsilon_{ev,w} \]

(25)

where \( \epsilon_{ev,w} \) is given by (14) with \( \epsilon_{ch} = \epsilon_{ch,w} \).

C. Examples

Figure 2 shows examples of the EXIT functions for check and variable nodes for a regular LDPC code with variable node degree \( d_v = 4 \) and check node degree \( d_c = 6 \). In Figure 2 we use \( I_{av} = 1 - h_b(\epsilon_{av}) \), \( I_{ev} = 1 - h_b(\epsilon_{ev}) \) and similarly for \( I_{ac} \) and \( I_{ec} \). The EXIT function for the hard decision channel (BSC) changes its behavior at certain values of \( I_{av} \).
These values correspond to a change of the majority decision rule of the Gallager B algorithm [3]. The EXIT function for the soft decision channel is an upper bound for all quantization schemes. As an example for larger output alphabets, we consider a binary symmetric quaternary output channel (BSQC) where the output of the channel takes on values from \{-D_{ch,2}, -D_{ch,1}, +D_{ch,1}, +D_{ch,2}\}. This channel output can be represented by a quantization using

$$\zeta = [0, \zeta_1, \infty].$$

Notice that for this case (22) gives \(p_0 = 0\). In Figure 2 the EXIT function is shown for \(\zeta_1 = 1.90\).

IV. IRREGULAR CODES

In this section we consider irregular LDPC codes, their EXIT functions, and constraints that have to be satisfied to avoid error floors. The results will be used in Section VII to optimize the degree distribution of the variable nodes. We focus on check-regular codes, i.e., all check nodes have the same degree \(d_c\), but the analysis extends in a simple way to check-irregular codes.

A. Mixing of EXIT Functions

We first prove the following theorem that is stated in a different context in [13]. Recall that \(g(l) = p_{L|X}(l + 1)\) is the conditional probability density of a channel from \(X\) to \(L\).

**Theorem 1.** Consider a collection of channels \(\{g_i(l)\}_i\) that satisfy \(g_i(-l) = e^l g_i(l)\) for all \(i\). We then have

$$\sum_i \lambda_i \ I(X_i; L_i) = I(X; L)$$

(27)

where

$$p_{L|X}(l + 1) = \sum_i \lambda_i \ p_{L|X}(l + 1) \cdot (l + 1).$$

(28)

In other words, for \(0 \leq \lambda_i \leq 1\) and \(\sum \lambda_i = 1\), the average EXIT function of the channel collection equals the EXIT function of the “averaged” channel.

**Proof:** Since \(g(l) = p_{L|X}(l + 1) = p_{L|X}(-l - 1)\), we obtain (note the integral limits)

$$I(X; L) = h(L) - h(L|X)$$

$$= \int_{-\infty}^{\infty} - [g(l) + g(-l)] \log \frac{g(l) + g(-l)}{2} + g(l) \log g(l) + g(-l) \log g(-l) \, dl.$$  

(29)

Using the symmetry condition \(g(-l) = e^{-l} g(l)\) this simplifies to

$$I(X; L) = \int_{-\infty}^{\infty} g(l) \left\{ e^{-l} \log \frac{2}{1 + e^l} + \log \frac{2}{1 + e^{-l}} \right\} \, dl$$

(30)

which is a linear operation on \(g(l)\). Hence the order of summation and computation of mutual information may be swapped.

Theorem 1 implies that the EXIT function of a mixture of codes can be computed as the average of the EXIT functions of the component codes as long as the channels to the \(L\)-values satisfy the property \(g(-l) = e^l g(l)\). In the case of binary message-passing decoders, however, the outgoing \(L\)-values of the variable node are quantized to \(\{+D_{ev}, -D_{ev}\}\). This nonlinear operation prohibits the exchange of averaging and the computation of the mutual information. On the other hand, the mixture EXIT function can still be computed by averaging over the crossover probabilities instead of the EXIT functions [1], [2], [14], i.e., we have

$$e_{ev} = \sum_i \lambda_i f_v(e_{av}; i, \epsilon_{ch})$$

(31)

where \(\lambda_i\) denotes the fraction of edges connected to a variable node of degree \(i\) [3] and \(f_v(\cdot)\) is given by [14]. Furthermore, we can formulate the successful decoding constraint in terms of crossover probabilities

$$e_{ev} = \sum_i \lambda_i f_v(e; i, \epsilon_{ch}) < f_c^{-1}(\epsilon)$$

(32)

for all \(\epsilon \in (0, 0.5)\). Expressing the design rate \(R\) of a code as

$$R = 1 - \delta,$$

(33)

leads to a linear program for maximizing the design rate:

maximize \(\sum_i \lambda_i\)

subject to \(\sum_i \lambda_i f_v(e; i, \epsilon_{ch}) < f_c^{-1}(\epsilon) \quad \forall \, \epsilon \in (0, 0.5)\)

\(\sum_i \lambda_i = 1\)

\(0 \leq \lambda_i \leq 1\).

In practice, the first constraint is evaluated for a fine grid of discrete \(\epsilon\). This linear program enables the efficient optimization of the variable node degree distribution.

B. Stability

It has been shown in [3] that if the degree distribution of an LDPC code satisfies a stability condition, the decoder converges to zero error when starting from a sufficiently small error probability. The stability condition for the binary message-passing decoder is given in the following theorem.

**Theorem 2.** An irregular LDPC code satisfies the stability condition under binary message-passing decoding (using quantized or unquantized channel messages) if and only if the variable node degree distribution satisfies

$$\left(\lambda_2 + 2 \epsilon_{ch} \lambda_3\right) (d_c - 1) < 1,$$

(35)

where \(\epsilon_{ch}\) denotes the error probability of a hard decision of the channel messages.

**Proof:** Let the superscript \((t)\) denote the error probabilities at iteration \(t\). Furthermore, let \(\epsilon_{ac}^{(t)} = \epsilon_{ev}^{(t)}\) and \(\epsilon_{av}^{(t+1)} = \epsilon_{ac}^{(t)}\) denote the error probabilities from the variable to check nodes and from the check to variable nodes, respectively. According
to (25), the EXIT function of the variable node decoder with quantized channel messages is the expectation of the EXIT functions of the sub-channels. Combining (6), (14) and (25) leads to

$$
\epsilon^{(t+1)}_{ev} = \sum_{w=0}^{W} p_w f_v \left( \epsilon^{(t)}_{av}; d_v, \epsilon_{ch,w} \right) 
= \sum_{w=0}^{W} p_w f_v \left( f_c(\epsilon^{(t)}_{ev}; d_c); d_v, \epsilon_{ch,w} \right). 
$$

(36)

For small error rates, it suffices to consider only the first order Taylor series expansion over one iteration [3]. Stability implies that the error probability decreases over one iteration for sufficiently small error probabilities, i.e., we have

$$
\lim_{\epsilon^{(t)}_{ev} \to 0} \frac{\partial \epsilon^{(t+1)}_{ev}}{\partial \epsilon^{(t)}_{ev}} = \lim_{\epsilon^{(t)}_{ev} \to 0} \left( \sum_{w=0}^{W} p_w \left. \frac{\partial f_v(\eta; d_v, \epsilon_{ch,w})}{\partial \eta} \right|_{\eta = f_v(\epsilon^{(t)}_{ev}; d_c)} \right) \cdot \lim_{\eta \to 0} \left( \frac{\partial f_c(\eta; d_c)}{\partial \eta} \right) < 1, 
$$

(37)

where the final step follows because $f_c(\epsilon^{(t)}_{ev}; d_c) \to 0$ as $\epsilon^{(t)}_{ev} \to 0$. For the check node (6), we have

$$
\lim_{\eta \to 0} \left( \frac{\partial f_c(\eta; d_c)}{\partial \eta} \right) = d_c - 1. 
$$

(38)

For the variable node decoder we start with the binomial cumulative distribution

$$
\lim_{p \to 0} \frac{\partial}{\partial p} B(k; n, p) = \lim_{p \to 0} \frac{\partial}{\partial p} \sum_{i=0}^{k} \binom{n}{i} p^i (1 - p)^{n-i} = \begin{cases} -n & ; k = 0 \\ 0 & ; k > 0 \end{cases}. 
$$

(39)

Using this result with (14) and (25) and the fact that for $\epsilon_{av} \to 0$ the reliability of the channel message $D_{ch}$ is small compared to the reliability of the a-priori message $D_{av}$, for regular LDPC codes we get

$$
\sum_{w=0}^{W} p_w \left( \lim_{\eta \to 0} \left. \frac{\partial f_v(\eta; d_v, \epsilon_{ch,w})}{\partial \eta} \right|_{\eta = f_v(\epsilon^{(t)}_{ev}; d_c)} \right) 
= \begin{cases} 1 & ; d_v = 2 \\ 2 \sum_{w=0}^{W} p_w & ; d_v = 3 \\ 0 & ; d_v \geq 4 \end{cases}. 
$$

(40)

The expectation of the error probabilities of all sub-channels equals the error probability of a hard decision channel $\epsilon_{ch} = \sum_{w=0}^{W} p_w \epsilon_{ch,w}$. Since every binary input symmetric output channel can be decomposed into BSCs [11], [12], [40] is also valid for unquantized channel messages by setting $\epsilon_{ch} = Q \left( \frac{\sigma_v}{2} \right)$. For irregular LDPC codes (31) leads to

$$
\sum_{i=1}^{W} \lambda_i \sum_{w=0}^{W} p_w \left( \lim_{\eta \to 0} \left. \frac{\partial f_v(\eta; i, \epsilon_{ch,w})}{\partial \eta} \right|_{\eta = f_v(\epsilon^{(t)}_{ev}; d_c)} \right) = \lambda_2 + 2 \epsilon_{ch} \lambda_3. 
$$

(41)

Combining (38) and (41) with (37) yields the theorem.

We note that this stability condition is the same as for Gallager’s original algorithm B (with hard decisions) as derived in [7], Eq. (11)), i.e. a finer quantization of the channel messages does not change the stability condition. The stability condition imposes a linear constraint on $\lambda_2$ and $\lambda_3$ and can hence be incorporated in the linear program (34) for code optimization.

For regular LDPC codes with $d_v = 2$, the left side of (35) is $d_c - 1$, which cannot be less than one. Therefore, such codes exhibit an error floor with binary message-passing decoding. Regular codes with $d_v = 3$ satisfy the stability condition if $2 \epsilon_{ch}(d_c - 1) < 1$. However, we show in the next section that these codes also exhibit an error floor with binary message-passing decoding.

C. Effect of Cycles

As shown in [15], cycles are unavoidable in the factor graph of finite length LDPC codes. Some cycles lead to error floors because they lead to poor properties of the code itself, e.g., cycles of length $g$ that include only variable nodes of degree two lead to a minimum distance $d_{min} \leq \frac{g}{2}$. For binary message-passing decoders, we identify cycles that lead to error floors that are caused by the decoding algorithm and not by the properties of the code.

Theorem 3. Consider a cycle formed by variable nodes of degree two and three only and assume that the channel messages associated with the nodes in the cycle are in error. If all other incoming messages at the check nodes of the cycle are correct, then the variable nodes forming the cycle cannot be corrected by the binary message-passing decoder.

Proof: An example of a cycle of interest is shown in Figure 3 where the left edges correspond to the channel
messages which are all assumed to be in error. In the first iteration the variable nodes send out their received messages. Since every check node in the cycle is connected twice to the set of erroneous variable nodes, the outgoing messages from the check nodes are also in error.

In the following iteration, the outgoing messages at the variable nodes are computed according to \( E \). For degree two variable nodes \((V_1 \text{ and } V_2)\), the extrinsic L-value of the sum of the channel L-value and the L-value of the other incoming message. Since both messages are in error, the outgoing message is also in error. For variable nodes of degree three, one message is not involved in the cycle (as shown for \( V_3 \) in Figure 3). Even if this message is correct, the extrinsic L-value is the sum of the channel L-value and two L-values from the check nodes with same magnitude but different signs. Therefore, the outgoing message is the sign of the channel L-value which is in error. Since this leads to the same state as after the first iteration, the decoder is not able to correct the errors of the variable nodes in the cycle.

If the factor graph contains cycles consisting of variable nodes of degree two and three only, there is a nonzero probability that the involved channel messages are in error leading to a decoding failure. Similar to stopping sets \([16]\), this situation leads to error floors. According to Theorem 3, to avoid an error floor, the factor graph of an LDPC code must not have such cycles. Such a graph exists if the following condition is satisfied.

**Theorem 4.** A factor graph with no cycles of variable nodes of degree two and three exists if and only if

\[
3\lambda_2 + 4\lambda_3 \leq \frac{6}{d_c} \left(1 - R\right) - \frac{1}{N} \leq \frac{6}{d_c}. \tag{42}
\]

**Proof:** Let \( \Lambda_i \) denote the fraction of variable nodes of degree \( i \), i.e. the node perspective of the variable node degree distribution \([3]\). Furthermore, let \( N \) and \( M \) denote the number of variable and check nodes, respectively. The maximum number of nodes in the subgraph containing only degree two and three variable nodes is

\[
\Lambda_2 N + \Lambda_3 N + M. \tag{43}
\]

This subgraph is cycle-free only if the number of edges \( E_t \) in the sub-graph is at most one less than the number of nodes

\[
E_t \leq \Lambda_2 N + \Lambda_3 N + M - 1. \tag{44}
\]

Furthermore, such a subgraph exists if \( \Lambda_1 \) is satisfied. Since \( E_t = 2\Lambda_2 N + 3\Lambda_3 N \), the bound \( \Lambda_1 \) is

\[
\Lambda_2 + 2\Lambda_3 \leq \frac{M - 1}{N}. \tag{45}
\]

Using

\[
\Lambda_i = \frac{\lambda_i}{\sum_j \lambda_j}, \tag{46}
\]

to convert from node perspective \( \Lambda_i \) to edge perspective \( \lambda_i \), and expressing the design rate \( R \) of the LDPC code as \( \frac{M}{N} \)

\[
R = 1 - \frac{1}{\sum_j \lambda_j} = 1 - \frac{M}{N} \tag{47}
\]

leads to the theorem.

An important consequence of Theorems 3 and 4 is that regular LDPC codes with \( d_v < 4 \) cannot be decoded without an error floor using binary message-passing decoders. To see this, observe that for regular codes with \( d_c = 2 \) or \( d_c = 3 \), Theorem 4 is satisfied only if \( d_v \leq 2 \) and \( d_v \leq 1.5 \), respectively. Both cases are not possible for codes of positive rate. Therefore, although regular LDPC codes with \( d_v = 3 \) are attractive from the point of view of a cycle-free analysis \([9]\), they have limited use for binary message-passing. To demonstrate this fact we show simulation results of such a code in Section V.

V. IMPLEMENTATION ASPECTS

A. Estimation of the A-Priori Channel

In Section III, we assumed that the variable node knows the parameters of the extrinsic channel for every iteration, i.e. it knows the crossover probabilities of the messages from the check nodes to the variable nodes. Suppose we represent these crossover probabilities as a sequence of numbers indexed by the iteration number. In \([17]\) we showed how this sequence can be predicted in advance from the trajectory of the decoder in the EXIT chart. Another possibility is to determine a sequence of crossover probabilities by simulations. We remark that for finite-length codes, such an approach can lead to better results than an asymptotic prediction from the EXIT chart. In this section, we present an adaptive method which is based on the fraction of unsatisfied check nodes \([18]\) and does not require a precomputed sequence of crossover probabilities.

Estimating the error probability based on the extrinsic messages of the check nodes is not possible, because these messages depend on the transmitted symbols which are not known to the decoder. However, the decoder knows that all \( d_c \) symbols involved in a parity-check equation sum up to zero, and it can therefore determine the extrinsic error probability \( \epsilon_{ec} \) from the number of unsatisfied check nodes denoted by \( M_c \). Denote the fraction of unsatisfied check nodes as

\[
\epsilon_s = \frac{M_c}{M}. \tag{48}
\]

This quantity can be seen as the extrinsic error probability of a check node of degree \( d_c + 1 \). Using \( \epsilon_s \), leads to an estimate of the crossover probability at the input of the check node

\[
\epsilon_{ac} \approx \frac{1 - (1 - 2\epsilon_s)^{2}}{2}. \tag{49}
\]

The crossover probability at the output of the check node follows from \( \epsilon_{ac} \) and \( \epsilon_{ec} \)

\[
\epsilon_{ec} = 1 - (1 - 2\epsilon_{ac})^{d_c - 1} \approx 1 - (1 - 2\epsilon_s)^{d_c - 1} \tag{50}
\]

This estimate is then used in \( \epsilon_{ac} \) to compute the reliability of the extrinsic channel.
We performed bit error rate simulations for codes of rate $R = 0.5$ for the BSC, BSQC and the soft information channel. The thresholds of these codes using the associated quantization schemes are shown in Table I where $\zeta_1$ is the quantization interval (see Section III-C) that leads to the best threshold. The bit error rate simulation results are shown in Figure 5 using codes of length $N = 10^4$. Also shown in this figure are the results for a regular LDPC code with variable node degree $d_v = 3$ and check node degree $d_c = 6$ transmitted over a channel with soft outputs. As expected from Section IV-C this code shows an error-floor.

B. Majority Decision Rules

The concept of converting all incoming messages to L-values, perform decoding in the L-value domain and sending the hard decision from variable to check nodes is useful for analyzing binary message-passing decoders. In practice, however, these operations are often replaced by majority decision rules due to reasons of complexity. Consider a variable node of degree $d_v$. Depending on $\epsilon_{av}$, a minimum number of $t$ messages (see (13)) from the check nodes have to disagree with the channel messages, in order to change the outgoing message of that variable node. This allows the derivation of a majority decision rule that is parametrized by $\epsilon_{av}$. Since $\epsilon_{av} = \epsilon_{ec}$ we can use the fraction of unsatisfied check nodes to adapt the majority decision rule over the iterations. For channels with larger output alphabet than binary, a majority decision rule has to be defined for every sub-channel (see [19]-[21]).

VI. NUMERICAL RESULTS

Using linear programming, we optimized codes using (34) with (35) and (32) and compared them with the capacity of the BIAWGN and BSC. For the optimization we set the maximum variable node degree to $d_{v,max} = 100$ and performed the optimization for check node degrees in the range between 2 and 1000. The decoding thresholds of these codes in terms of the required signal to noise ratio are shown in Figure 4. Note that cycles of the kind described in Theorem 3 could occur if a random interleaver was used. However, we constructed our codes using the progressive edge-growth (PEG) algorithm [19], [20] which ensures that such cycles do not exist.

Observe that the gap to capacity decreases as the rate increases. This makes binary message-passing decoders attractive for applications which require a high code rate. For a rate of 0.9, the best code using soft channel information is within approximately 0.5 dB of capacity. Note that the soft-decision and hard-decision channel EXIT curves serve as upper and lower bounds, respectively, for all quantization schemes.

Table I Thresholds for Code of Rate 0.5 and the LDPC Code Used in [4, Sec. I.6].

The system with hard channel decisions (BSC) corresponds to the Gallager B algorithm. Observe that by adding one more bit for the channel messages and quantizing them according to a BSQC, the performance of this algorithm can be improved by 1.0 dB with only a small increase in decoding complexity. A finer quantization of the channel messages will not result in a significant gain, since the gap to the unquantized system is only approximately 0.5 dB. Remark that predicting the sequence of crossover probabilities by using the trajectory in the EXIT chart [17] gives a similar performance.

To demonstrate the performance of binary message-passing decoders for high code rates we perform bit error rate simulations of a code used in [4, Section I.6.2] which is a regular LDPC code with variable node degree $d_v = 7$, check node degree $d_c = 112$ and is used for an optical transmission system for high data rates up to 40 Gbps. The slopes which define the code (for details see [4, Section I.6.2]) are not defined in the standard and have been chosen to be the first seven prime numbers.
i.e., 2, 3, 5, 7, 11, 13, 17. In [4] it is assumed that the decoder observes hard decisions from the channel. Using our analysis, the thresholds for channels with larger output alphabets are shown in Table [4] assuming that the unquantized channel can be modeled as a BIAWGN channel. The corresponding bit error rate simulations are shown in Figure [5]. Increasing the number of quantization levels of the channel messages by one bit leads to an improvement of approximately 1.0 dB.

![Figure 6](image_url)

Fig. 6. Bit error rate simulations for the LDPC code of rate $R = 0.9375$ used in [4] Sec. I.6.

VII. CONCLUSIONS

We analyzed binary message-passing decoders using EXIT charts. For channels which deliver hard decisions, this analysis led to an algorithm that is equivalent to Gallager’s decoding algorithm B. We extended these results to channels with larger output alphabets including channels providing soft information. We found that increasing the channel output alphabet size by only one bit leads to a significant lowering of the decoding threshold. We described why the mixing property of EXIT functions does not apply to binary message-passing algorithms if one uses mutual information, and presented a modified mixing method based on error probabilities in order to optimize codes. Finally, we showed that degree two and three variable nodes forming a cycle cannot be corrected and we derived a condition for the variable node distribution that guarantees the existence of a cycle-free subgraph for these nodes.

REFERENCES

[1] R. Gallager, “Low density parity check codes,” IRE Transactions on Information Theory, vol. IT-8, pp. 21–28, Jan. 1962.
[2] ———, Low Density Parity Check Codes, ser. Research monograph series. Cambridge, Mass.: MIT Press, 1963, no. 21.
[3] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” Information Theory, IEEE Transactions on, vol. 47, pp. 599 – 618, Feb. 2001.
[4] ITU-T G.975.1, “Forward error correction for high bit-rate DWDM submarine systems,” 2004.
[5] S. ten Brink, “Convergence of iterative decoding,” Electronics Letters, vol. 35, no. 10, pp. 806–808, May 1999.
[6] A. Ashikhmin, G. Kramer, and S. ten Brink, “Extrinsic information transfer functions: model and erasure channel properties,” Information Theory, IEEE Transactions on, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.
[7] M. Ardakani and F. Kschischang, “Properties of optimum binary message-passing decoders,” Information Theory, IEEE Transactions on, vol. 51, no. 10, pp. 3658–3665, Oct. 2005.
[8] M. Ardakani, “Efficient analysis, design and decoding of low-density parity-check codes,” Ph.D. dissertation, University of Toronto, 2004.
[9] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” Information Theory, IEEE Transactions on, vol. 42, no. 2, pp. 429–445, Mar. 1996.
[10] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” Information Theory, IEEE Transactions on, vol. 47, pp. 619–637, Feb. 2001.
[11] I. Land, “Reliability information in channel decoding – practical aspects and information theoretical bounds,” Ph.D. dissertation, University of Kiel, Germany, 2005.
[12] I. Land and J. Huber, “Information combining,” Foundations and Trends in Commun. and Inform. Theory, vol. 3, no. 3, 2006.
[13] M. Tucery and J. Hagenauer, “EXIT charts of irregular codes,” in 2002 Conference on Information Sciences and Systems, Princeton University, 2002.
[14] M. Ardakani and F. Kschischang, “Designing irregular LDPC codes using EXIT charts based on message error rate,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Lausanne, Switzerland, 2002.
[15] T. Etzion, A. Trachtenberg, and A. Vardy, “Which codes have cycle-free Tanner graphs?” Information Theory, IEEE Transactions on, vol. 45, no. 6, pp. 2173–2180, Sep. 1999.
[16] T. Richardson and R. Urbanke, Modern Coding Theory. Cambridge University Press, 2008.
[17] G. Lechner, T. Pedersen, and G. Kramer, “EXIT chart analysis of binary message-passing decoders,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Nice, France, 2007.
[18] G. Yue and X. Wang, “A new binary iterative LDPC decoding algorithm,” in Proc. Int. Symp. on Turbo Codes & Rel. Topics, Lausanne, Switzerland, 2008.
[19] X.-Y. Hu, E. Eleftheriou, and D. Arnold, “Regular and irregular progressive edge-growth Tanner graphs.” Information Theory, IEEE Transactions on, vol. 51, no. 1, pp. 386–398, Jan. 2005.
[20] X.-Y. Hu, “Software for PEG code construction.” [Online]. Available: http://www.inference.phy.cam.ac.uk