Exchange of Limits: Why Iterative Decoding Works

Satish Babu Korada and Rüdiger Urbanke
School of Computer and Communication Science
EPFL, Lausanne CH-1015, Switzerland
Email: {satish.korada, ruediger.urbanke}@epfl.ch

***Abstract*— We consider communication over binary-input memoryless output-symmetric channels using low-density parity-check codes under MP decoding. The asymptotic (in the length) performance of such a combination for a fixed number of iterations is given by density evolution. It is customary to define the *threshold* of density evolution as the maximum channel parameter for which the bit error probability under density evolution converges to zero as a function of the iteration number.**

In practice we often work with short codes and perform a large number of iterations. It is therefore interesting to consider what happens if in the standard analysis we exchange the order in which the blocklength and the number of iterations diverge to infinity. In particular, we can ask whether both limits give the same threshold.

Although empirical observations strongly suggest that the exchange of limits is valid for *all* channel parameters, we limit our discussion to channel parameters below the density evolution threshold. Specifically, we show that under some suitable technical conditions the bit error probability vanishes below the density evolution threshold regardless of how the limit is taken.

I. Introduction

A. Motivation

Consider transmission over a binary-input memoryless output-symmetric (BMS) channel using a low-density parity-check (LDPC) code and decoding via a message-passing (MP) algorithm. We refer the reader to [1] for an introduction to the standard notation and an overview of the known results. It is well known that, for good choices of the degree distribution and the MP decoder, one can achieve rates close to the capacity of the channel with low decoding complexity [2].

The standard analysis of iterative decoding systems assumes that the blocklength 'n' is large (tending to infinity) and that a fixed number of iterations is performed. As a consequence, when decoding a given bit, the output of the decoder only depends on a fixed-size local neighborhood of this bit and this local neighborhood is tree-like. This local tree property implies that the messages arriving at nodes are conditionally independent, significantly simplifying the analysis. To determine the performance in this setting, we track the evolution of the message-densities as a function of the iteration. This process is called *density evolution* (DE). Denote the probability of bit error of a code $\mathtt{G}$ after ℓ iterations by $P_b(\mathtt{G}, \epsilon, \ell)$, where ϵ is the channel parameter. Then DE computes $\lim_{n\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$. If we now perform more and more iterations then we get a limiting performance corresponding to

$$\lim_{\ell\to\infty} \lim_{n\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]. \tag{1}$$

A necessary condition for the computation graphs of depth ℓ to all nodes form trees is that the number of iterations does not exceed $c\log(n)$, where c is a constant that only depends on the degree distribution. (For a $(\mathtt{l}, \mathtt{r})$-regular degree distribution pair a valid choice of c is $c(\mathtt{l}, \mathtt{r}) = \frac{2}{\log(\mathtt{l}-1)(\mathtt{r}-1)}$, [3].) In practice, this condition is rarely fulfilled: standard blocklengths measure only in the hundreds or thousands but the number of iterations that have been observed to be useful in practice can easily exceed one hundred.

Consider therefore the situation where we fix the blocklength but let the number of iterations tend to infinity, i.e., we consider the limit $\lim_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$. Now take the blocklength to infinity, i.e., consider

$$\lim_{n\to\infty} \lim_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]. \tag{2}$$

What can we say about (2) and its relationship to (1)?

Consider the belief propagation (BP) algorithm. It was shown by McEliece, Rodemich, and Cheng [4] that one can construct specific graphs and noise realizations so that the messages on a specific edge either show a chaotic behavior or converge to limit cycles. In particular, this means that the messages do not converge as a function of the iteration. For a fixed length and a discrete channel, the number of graphs and noise realizations is finite. Therefore, if for single graph and noise realization the messages do not converge as a function of ℓ, then it is likely that also $\lim_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$ does not converge as a function of ℓ (unless by some miracle the various non-converging parts cancel). Let us therefore consider $\limsup_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$ and $\liminf_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$. What happens if we increase the blocklength and consider $\lim_{n\to\infty} \limsup_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$ and $\lim_{n\to\infty} \liminf_{\ell\to\infty} \mathbb{E}[P_b(\mathtt{G}, \epsilon, \ell)]$?

Assume that the given combination (of the channel family and the decoder MP) has a threshold in the following sense: for the given channel family characterized by the real valued parameter ϵ there exists a value ϵ^{MP} so that for all $0 \le \epsilon < \epsilon^{\text{MP}}$ the DE limit (1) is 0, whereas for all $\epsilon > \epsilon^{\text{MP}}$ it is strictly positive. Although empirical observations strongly suggest that the exchange of limits is valid for *all* channel parameters ϵ, we limit our discussion to channel parameters below the DE threshold ϵ^{MP}. In this case DE promises bit error probabilities that tend to zero.

Instead of considering the simple exchange of limits one can consider joint limits where the iteration is an arbitrary but increasing function of the blocklength, i.e., one can consider $\lim_{n\to\infty} \mathbb{E}[P_b^{\text{MP}}(\mathtt{G}, \epsilon, \ell(n))]$. Although our arguments extend

to this case, for the sake of simplicity we restrict ourselves to the standard exchange of limits discussed above. Further, we restrict ourselves to regular ensembles. All the difficulties encountered in the analysis are already contained in this case.

B. Definition and Notations

Consider a MP algorithm with message alphabet $\mathcal{M}$. Assume that the algorithm is symmetric in the sense of [1][Definition 4.81, p. 210], so that for the purpose of analysis it is sufficient to restrict our attention to the all-one codeword assumption.

The tools we develop can be applied to a variety of MP decoders. To be concrete, we discuss below a few interesting examples. In the following, by reliability of a message μ we mean its absolute value $|\mu|$. This means that the message $-\mu$ and μ have the same reliability.

Definition 1 (Bounded MS, BP Decoders): The *bounded* min-sum (MS(M)) decoder and *bounded* belief propagation (BP(M)) decoder, both with parameter $M \in \mathbb{R}^+$, are identical to the standard min-sum and belief propagation decoder except that the reliability of the messages emitted by the check nodes is bounded to M before the messages are forwarded to the variable nodes. $\Diamond$

II. Sufficient Conditions Based on Expansion Arguments

Let us now show that for codes with sufficient expansion the exchange of limits is indeed valid below the DE decoding threshold.

Burshtein and Miller were the first to realize that expansion arguments can be applied not only to the flipping algorithm but also to show that certain MP algorithms have a fixed error correcting radius [5]. Although their results can be applied directly to our problem, we get somewhat stronger statements by using the expansion in a slightly different manner.

The advantage of using expansion is that the argument applies to a wide variety of decoders and ensembles. On the negative side, the argument can only be applied to ensembles with large left degree. Why do we need large left degrees to prove the result? There are two reasons why a message emitted by a variable node can be bad (let bad mean incorrect). This can be due to the received value, or it can be due to a large number of bad incoming messages. If the degree of the variable node is large then the received value plays only a minor role (think of a node of degree 1000; in this case the received value has only a limited influence on the outgoing message and this message is mostly determined by the 999 incoming messages). Suppose that the left degree is large and ignore therefore for a moment the received message. In this case large expansion helps for the following reason.

Consider a fixed iteration ℓ. Let $\mathcal{B}_\ell$ denote the set of bad variable nodes in iteration ℓ (the set of variable nodes that emit bad messages in iteration ℓ). Perform one further round of MP. In the next iteration the only check nodes which send bad messages are those connected to $\mathcal{B}_\ell$. Therefore, for a variable to belong to $\mathcal{B}_{\ell+1}$, it must be connected to a large number of bad check nodes, and hence must share many check-node neighbors with variables in $\mathcal{B}_\ell$. Suppose that $\mathcal{B}_\ell$ and $\mathcal{B}_{\ell+1}$ are sufficiently small and that the graph has large expansion. Then the number of common check-node neighbors of $\mathcal{B}_\ell$ and $\mathcal{B}_{\ell+1}$ can not be too large (since otherwise the expansion would be violated). This limits the maximum relative size of $\mathcal{B}_{\ell+1}$ with respect to $\mathcal{B}_\ell$. In other words, once $\mathcal{B}_\ell$ has reached a sufficiently small size (so that the expansion arguments can be applied), the number of errors quickly converges to zero with further iterations. In order to achieve good bounds the above argument has to be refined, but it does contain the basic idea of why large expansion helps.

On the other hand, if variable nodes have small degrees, then the received values play a dominant role and can no longer be ignored. As a consequence, for small degrees expansion arguments no longer suffice by themselves.

Why are we using expansion arguments if we are interested in standard LDPC ensembles? It is well known that such codes are good expanders with high probability [5]. More precisely, we say that a $(\mathtt{l},\mathtt{r})$ bipartite graph is an $(\mathtt{l},\mathtt{r},\alpha,\gamma)$-left expander if all variable node sets $\mathcal{V}$ of size $|\mathcal{V}| \leq \alpha n$ have at least $\gamma \mathtt{l} |\mathcal{V}|$ check-node neighbors. It is not hard to see that γ can not be larger than $1-\frac{1}{\mathtt{l}}$; take a check node and draw its computation graph of height ℓ. Let $\mathcal{V}$ be the set of variable nodes contained in this subgraph. For $\ell = 1$ this subgraph contains $1+\mathtt{r}(\mathtt{l}-1)$ check nodes and and $\mathtt{r}$ variable nodes. For depth ℓ, the number of check and variables nodes are $\frac{\mathtt{r}(\mathtt{l}-1)^{\ell+1}(\mathtt{r}-1)^{\ell}-\mathtt{l}}{\mathtt{l}\mathtt{r}-\mathtt{l}-\mathtt{r}}$ and $\frac{\mathtt{r}(\mathtt{l}-1)^{\ell}(\mathtt{r}-1)^{\ell}-\mathtt{r}}{\mathtt{l}\mathtt{r}-\mathtt{l}-\mathtt{r}}$. The expansion of such a subgraph is at most $\frac{1}{\mathtt{l}}\frac{\mathtt{r}(\mathtt{l}-1)^{\ell+1}(\mathtt{r}-1)^{\ell}-\mathtt{l}}{\mathtt{r}(\mathtt{l}-1)^{\ell}(\mathtt{r}-1)^{\ell}-\mathtt{r}}$ and it rapidly converges to $1-\frac{1}{\mathtt{l}}$ by choosing ℓ larger and larger. Surprisingly, for any $\gamma < 1-\frac{1}{\mathtt{l}}$, there exists an $\alpha(\gamma) > 0$, such that for sufficiently large n with high probability a random graph is an $(\mathtt{l},\mathtt{r},\alpha,\gamma)$-left expander.

Let us start with ensembles that have large variable degrees. The key to what follows is to find a proper definition of a "good" pair of message subsets.

Definition 2 (Good Message Subsets): For a fixed $(\mathtt{l},\mathtt{r})$-regular ensemble and a fixed MP decoder, let β, $0 < \beta \leq 1$, be such that $\beta(\mathtt{l}-1) \in \mathbb{N}$. A "good" pair of subsets of $\mathcal{M}$ of "strength" β is a pair of subsets $(G_{\mathtt{v}}, G_{\mathtt{c}})$ so that

- if $\beta(l-1)$ of the $(\mathtt{l}-1)$ incoming messages at a variable node belong to $G_{\mathtt{v}}$ then the outgoing message on the remaining *edge* is in $G_{\mathtt{c}}$
- if all the $(\mathtt{r}-1)$ incoming messages at a check node belong to $G_{\mathtt{c}}$ then the outgoing message on the remaining *edge* is in $G_{\mathtt{v}}$
- if $\beta(\mathtt{l}-1)+1$ of all $\mathtt{l}$ incoming messages belong to $G_{\mathtt{v}}$, then the *variable* is decoded correctly

We denote the probability of the bad message set $\mathcal{M}\backslash G_{\mathtt{v}}$ after ℓ iterations of DE by $p_{\text{bad}}^{(\ell)}$. $\Diamond$

As we will see shortly, for most of the decoders the sets $G_{\mathtt{v}}$ and $G_{\mathtt{c}}$ can be chosen to be equal (but the BP(M) decoder is an interesting case where $G_{\mathtt{v}} \neq G_{\mathtt{c}}$).

Theorem 3 (Expansion and Bit Error Probability): Consider an LDPC$(n,\mathtt{l},\mathtt{r})$ ensemble, transmission over a

BMS(ϵ) channel, and a symmetric MP decoder. Assume that this combination has a threshold under DE, call it ϵ^{MP}. Let β be the strength of the good message subset. If $\beta < 1$ and if for some $\epsilon < \epsilon^{\text{MP}}$ we have $p_{\text{bad}}^{(\infty)} = 0$ then

$$\lim_{n\to\infty} \limsup_{\ell\to\infty} \mathbb{E}_{\text{LDPC}(n,\mathtt{l},\mathtt{r})}[P_b^{\text{MP}}(\mathtt{G},\epsilon,\ell)] = 0. \tag{3}$$

The proof idea is somewhat different from the one used in [5]. We first perform a small number of iterations to bring the error probability down to a small value. But rather than asking that the error probability decreases to zero by performing a sufficient number of further iterations, we only require that it stays small. The payoff for this less stringent requirement is that the necessary conditions are less stringent as well. The following theorem is more in the spirit of [5].

Theorem 4 (Expansion and Block Error Probability): Consider an LDPC$(n,\mathtt{l},\mathtt{r})$ ensemble, transmission over a BMS(ϵ) channel, and a symmetric MP decoder. Assume that this combination has a threshold under DE, call it ϵ^{MP}. Let β be the strength of the good message subset. If $\beta < \frac{\mathtt{l}-2}{\mathtt{l}-1}$ and if for some $\epsilon < \epsilon^{\text{MP}}$ we have $p_{\text{bad}}^{(\infty)} = 0$ then

$$\lim_{n\to\infty} \limsup_{\ell\to\infty} \mathbb{E}_{\text{LDPC}(n,\mathtt{l},\mathtt{r})}[P_B^{\text{MP}}(\mathtt{G},\epsilon,\ell)] = 0. \tag{4}$$

As in Theorem 3 we first perform a fixed number of iterations to bring down the bit error probability below a desired level. We then use Theorem 5, a modified version of a theorem by Burshtein and Miller [5], to show that for a graph with sufficient expansion the MP algorithm decodes the whole block correctly once the bit error probability is sufficiently small.

Theorem 5 ([5]): Consider an $(\mathtt{l},\mathtt{r},\alpha,\gamma)$-left expander. Assume that $0 \le \beta \le 1$ such that $\beta(l-1) \in \mathbb{N}$ and that $\beta\frac{\mathtt{l}-1}{\mathtt{l}} \le 2\gamma - 1$. Let $n_0 \le \frac{\alpha}{\mathtt{l}\mathtt{r}}n$. If at some iteration ℓ the number of bad variable nodes is less than n_0 then the MP algorithm will decode successfully.

Discussion: Theorem 4 has a stronger implication (the block error probability tends to zero as a function of the iteration, assuming the bit error probability has reached a sufficiently small value) than Theorem 3 (here we are only guaranteed that the bit error probability stays small once it has reached a sufficiently small value). But it also requires a considerably stronger condition.

Let us now apply the previous theorems to some examples.

Example 6 (BSC and GalB Algorithm): For this algorithm $\mathcal{M} = \{-1,+1\}$. Pick $G_{\mathtt{v}} = G_{\mathtt{c}} = \{+1\}$. Assume that the received value (via the channel) is incorrect. In this case at least $\lceil(\mathtt{l}-1)/2\rceil + 1$ of the $(\mathtt{l}-1)$ incoming messages should be good to ensure that the outgoing message is good and at least $\lceil(\mathtt{l}-1)/2\rceil + 2$ of the $\mathtt{l}$ incoming messages should be good to ensure that the variable is decoded correctly. Therefore, $\beta = \frac{\lceil(\mathtt{l}-1)/2\rceil+1}{\mathtt{l}-1}$. If the probability of the bad message set goes to 0 in the DE limit, then from Theorem 3 the limits can be exchanged if $\mathtt{l}-1 > 1+\lceil(\mathtt{l}-1)/2\rceil$, i.e., for $\mathtt{l} \ge 5$ and from Theorem 4, the block error probability goes to zero if $\mathtt{l}-2 > 1+\lceil(\mathtt{l}-1)/2\rceil$, i.e., for $\mathtt{l} \ge 7$. $\Diamond$

The key to applying expansion arguments to decoders with a continuous alphabet is to ensure that the received values are no longer dominant once DE has reached small error probabilities. This can be achieved by ensuring that the input alphabet is smaller than the message alphabet. Let us give a few examples here.

Example 7 (MS(5) Decoder): Consider $(\mathtt{l} \ge 5, \mathtt{r})$ code and fix $M = 5$. Let the channel log likelihoods belong to $[-1,1]$. It is easy to check that in this case we can choose $G_{\mathtt{v}} = G_{\mathtt{c}} = [4,5]$ and that it has strength $\beta \le \frac{3}{4}$. Therefore, if the probability of the bad message set goes to 0 under DE, then according to Theorem 3 the limits can be exchanged. If instead we consider $(\mathtt{l} \ge 7, \mathtt{r})$ then $\beta \le \frac{1}{3}$. Hence, according to Theorem 4 the block error probability tends to 0. $\Diamond$

Example 8 (BP(10) Decoder): Let $\mathtt{l} = 5$ and $\mathtt{r} = 6$ and fix $M = 10$. Let the channel log likelihoods belong to $[-1,1]$. We claim that in this case the message subset pair $G_{\mathtt{v}} = [9,10], G_{\mathtt{c}} = [16,41]$ is good with strength $\beta = \frac{3}{4}$. This can be seen as follows: If all the incoming messages to a check node belong to $G_{\mathtt{c}}$, then the outgoing message is at least 14.39, which is mapped down to 10. Suppose that at a variable node at least $3(= \beta(\mathtt{l}-1))$ out of the 4 incoming messages belong to $G_{\mathtt{v}}$. In this case the reliability of the outgoing message is at least $16 = 3\times 9 - 10 - 1$. The maximum reliability is 41. Moreover, if all the incoming messages belong to $G_{\mathtt{v}}$ then the variable is decoded correctly. Therefore if the probability of outgoing messages from check nodes being in $[9,10]$ goes to 1 in the DE limit then from Theorem 3, the limits can be exchanged. $\Diamond$

It is clear that Theorems 3 and 4 apply to an infinite variety of decoders. But in all these cases the required variable node degrees are rather large. In the next section we discuss an alternative method which can sometimes be applied to ensembles with low variable-node degrees.

III. Sufficient Condition Based on Birth-Death Process

A. Main Result and Outline

As we have mentioned before, if the left degree is small then the received value retains a large influence on emitted messages regardless of the number of iterations. In this case expansion arguments no longer suffice to prove our desired result. As a representative example let us therefore consider the case of $\mathtt{l} = 3$. Although the results below can be extended to more general scenarios, we limit the subsequent discussion to the Gallager decoding algorithm B (GalB). All the complications are already present for this case.

Lemma 9 (Exchange of Limits): Consider transmission over the BSC(ϵ) using random elements from the $(\mathtt{l}=3,\mathtt{r})$-regular ensemble and decoding by the GalB algorithm. If $\epsilon < \epsilon^{\text{LGalB}}$ then

$$\lim_{n\to\infty} \limsup_{\ell\to\infty} \mathbb{E}[P_b^{\text{GalB}}(\mathtt{G},\epsilon,\ell)] = 0,$$

where ϵ^{LGalB} is the smallest parameter ϵ for which a solution to the fixed point equation

$$x = \bar{\epsilon}(1-(1-x)^{\mathtt{r}-1})^2 + \epsilon(1-(1-x)^{2(\mathtt{r}-1)})$$

$\mathtt{r}$	rate	ϵ^{Sha}	ϵ^{GalB}	ϵ^{LGalB}
4	0.25	≈ 0.2145	≈ 0.1068	≈ 0.0847
5	0.4	≈ 0.1461	≈ 0.06119	≈ 0.0506
6	0.5	≈ 0.11002	≈ 0.0394	≈ 0.0336

TABLE I

THRESHOLD VALUES FOR SOME DEGREE DISTRIBUTIONS.

exists in $(0, \epsilon]$.

Example 10: Table I shows thresholds for $\mathtt{r} = 4, 5, 6$. For the $(\mathtt{l} = 3, \mathtt{r} = 6)$ degree distribution we have $\epsilon^{\text{LGalB}} \approx 0.0336$. This is slightly smaller than, but comparable to, $\epsilon^{\text{GalB}} \approx 0.0394$. $\Diamond$

Due to space constraints we do not present the proof in detail. But we will discuss the ideas behind the main steps.

B. All-One Codeword Assumption

Fix $0 \leq \epsilon < \epsilon^{\text{LGalB}}$. We prove that for every $\alpha > 0$ there exists an $n(\alpha, \epsilon)$ so that $\limsup_{\ell \to \infty} \mathbb{E}[P_b^{\text{GalB}}(\mathtt{G}, \epsilon, \ell)] < \alpha$ for $n \geq n(\alpha, \epsilon)$. Without loss of generality we can assume that the all-one codeword was sent. Therefore, the message 1 signifies in the sequel a *correct* message, whereas -1 implies that the message is *incorrect*.

C. Linearized Gal B

The analysis is simplified considerably by *linearizing* the decoding algorithm in the following way. Define the *Linearized Gallager B* (LGalB) algorithm. The LGalB algorithm has the same processing rules at the variable nodes as the GalB algorithm. At check nodes, however, an outgoing message is -1 (incorrect) if *any* of the incoming messages is -1 (incorrect). It is not difficult to check that the error probability of LGalB is an upper bound on the error probability for GalB. Note that ϵ^{LGalB} as given in Lemma 9 is the DE threshold corresponding to LGalB.

We will prove that for every $0 \leq \epsilon < \epsilon^{\text{LGalB}}$ and every $\alpha > 0$ there exists an $n(\alpha, \epsilon)$ so that $\limsup_{\ell \to \infty} \mathbb{E}[P_b^{\text{LGalB}}(\mathtt{G}, \epsilon, \ell)] < \alpha$ for $n \geq n(\alpha, \epsilon)$.

D. Marking Process

The *marking* process allows us (i) to consider an *asynchronous* version of LGalB (i.e., the schedule of the computation is no longer important) and (ii) ensures that we are dealing with a monotone increasing function.

More precisely, we split the process into two phases: we start with LGalB for $\ell(p)$ iterations to get the error probability below p; we then continue the marking process associated with an infinite number of further iterations of LGalB. This means that we mark any variable that is bad in at least one iteration $\ell \geq \ell(p)$. Clearly, the union of all variables that are bad at at least one point in time $\ell \geq \ell(p)$ is an upper bound on the maximum number of variables that are bad at any specific instance in time.

The standard *schedule* of the LGalB is parallel, i.e., all incoming messages (at either variable or check nodes) are processed at the same time. This is the natural schedule for an actual implementation. For the purpose of analysis it is convenient to consider an *asynchronous* schedule.

For a given graph $\mathtt{G}$, and channel realization $\mathtt{E}$, let $\mathcal{M}(\mathtt{G}, \mathtt{E}, \ell)$ denote the set of marked variables at the end of the process assuming that the initial set of marked edges is the set of bad edges after ℓ rounds of LGalB. Let $M(\mathtt{G}, \mathtt{E}, \ell) = |\mathcal{M}(\mathtt{G}, \mathtt{E}, \ell)|$. It is not hard to see that for any $\ell' \geq \ell$, $P_b^{\text{LGalB}}(\mathtt{G}, \epsilon, \ell') \leq \mathbb{E}_{\mathtt{E}}[M(\mathtt{G}, \mathtt{E}, \ell)]/n$.

E. Witness

It remains to bound $\mathbb{E}[M(\mathtt{G}, \mathtt{E}, \ell)]$. The difficulty in analyzing the marking process lies in the fact that after $\ell(p)$ iterations the set of starting edges for the marking process depends on the noise realization as well as the graph. Our aim therefore is to reduce this correlated case to the uncorrelated case by a sequence of transformations. As a first step we show how to get rid of the correlation with respect to the noise realization.

Consider a fixed graph $\mathtt{G}$. Assume that we have performed ℓ iterations of LGalB. For each edge $\mathtt{e}$ that is bad in the ℓ-th iteration we construct a "witness." A witness for $\mathtt{e}$ is a subset of the computation tree of height ℓ for $\mathtt{e}$ consisting of paths that carry bad messages. We construct the witness recursively starting with $\mathtt{e}$. Orient $\mathtt{e}$ from check node to variable node. At any point in time while constructing the witness associated to $\mathtt{e}$ we have a partial witness that is a tree with oriented edges. The initial such partial witness is $\mathtt{e}$. One step in the construction consists of taking a leaf edge of the partial witness and to "grow it out" according to the following rules.

If an edge enters a variable node that has an incorrect received value then add the *smallest* (according to some fixed but arbitrary order on the set of edges) edge that carries an incorrect incoming message to the witness and continue the process along this edge. The added edge is directed from variable node to check node. If an edge enters a variable node that has a correct received value then add both incoming edges to the witness and follow the process along both edges. (Note that in this case both of these edges must have carried bad messages.) Again, both of these edges are directed from variable to check node. If an edge enters a check node then choose the smallest incoming edge that carries an incorrect message and add it to the witness. Continue the process along this edge. The added edge is directed from check to variable node. Continue the process until depth ℓ. Fig. 1 shows an example for $\mathtt{l} = 3$, $\mathtt{r} = 4$, and $\ell = 2$. Denote the union of all witnesses for all edges that are bad in the ℓ-th iteration by $\mathcal{W}(\mathtt{G}, \mathtt{E}, \ell)$. We simply call it *the witness*. The witness is a part of the graph that on its own explains why the set of bad edges after ℓ iterations is bad.

How large is $\mathcal{W}$? The larger ℓ, the fewer bad edges we expect to see in iteration ℓ. On the other hand, the size of the witness for each bad edge grows as a function of ℓ. Fortunately one can show that the first effect dominates and that the size of the witness vanishes as a function of the iteration number.

F. Randomization

A witness $\mathcal{W}$ consists of two parts, (i) the graph structure of $\mathcal{W}$ and (ii) the channel realizations of the variables in $\mathcal{W}$.

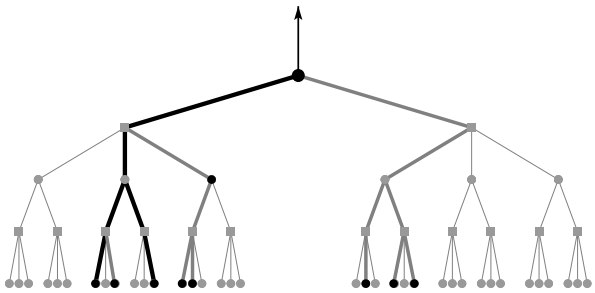

Fig. 1. Construction of the witness for a bad edge. The *dark* variables represent channel errors. The part of the tree with *dark* edges represent the witness, the *thick* edges, including both dark and grey, represent the bad messages in the past iterations.

By some abuse of notation we write $\mathcal{W}$ also if we refer only to the graph structure or only to the channel realizations.

Fix a graph $\mathtt{G}$ and a witness $\mathcal{W}$, $\mathcal{W} \subseteq \mathtt{G}$. Let $\mathcal{E}_{\mathtt{G},\mathcal{W}}$ denote the set of all error realizations $\mathtt{E}$ that give rise to $\mathcal{W}$, i.e., $\mathcal{W}(\mathtt{G}, \mathtt{E}, \ell) = \mathcal{W}$. Clearly, for all $\mathtt{E} \in \mathcal{E}_{\mathtt{G},\mathcal{W}}$ we must have $\mathcal{W} \subseteq \mathtt{E}$. In words, on the set of variables fixed by the witness the errors are fixed by the witness itself. Therefore, the various $\mathtt{E}$ that create this witness differ only on $\mathtt{G}\backslash\mathcal{W}$. As a convention, we define $\mathcal{E}_{\mathtt{G},\mathcal{W}} = \emptyset$ if $\mathcal{W} \not\subseteq \mathtt{G}$.

Let $\mathcal{E}'_{\mathtt{G},\mathcal{W}}$ denote the set of projections of $\mathcal{E}_{\mathtt{G},\mathcal{W}}$ onto the variables in $\mathtt{G}\backslash\mathcal{W}$. Let $\mathtt{E}' \in \mathcal{E}'_{\mathtt{G},\mathcal{W}}$. Think of $\mathtt{E}'$ as an element of $\{0,1\}^{|\mathtt{G}\backslash\mathcal{W}|}$, where 0 denotes a correct received value and 1 denotes an incorrect received value. In this way, $\mathcal{E}'_{\mathtt{G},\mathcal{W}}$ is a subset of $\{0,1\}^{|\mathtt{G}\backslash\mathcal{W}|}$.

This is important: $\mathcal{E}'_{\mathtt{G},\mathcal{W}}$ has structure. We claim that, if $\mathtt{E}' \in \mathcal{E}'_{\mathtt{G},\mathcal{W}}$ then $\mathcal{E}'_{\mathtt{G},\mathcal{W}}$ also contains $\mathtt{E}'_{\leq}$, i.e., it contains all elements of $\{0,1\}$ that are smaller than $\mathtt{E}'$ with respect to the natural partial order on $\{0,1\}^{|\mathtt{G}\backslash\mathcal{W}|}$. More precisely, if the noise realization $\mathtt{E}' \in \mathcal{E}'_{\mathtt{G},\mathcal{W}}$ gives rise to the witness $\mathcal{W}$ then converting any incorrect received value in $\mathtt{E}'$ to a correct one will also give rise to $\mathcal{W}$. The proof of the following lemma relies heavily on this property. By some abuse of notation, let $\mathcal{M}(\mathtt{G}, \mathtt{E}, \mathcal{W})$, be the marking process with the edges in $\mathcal{W}$ as the initial set of bad edges.

Lemma 11 (Channel Randomization): Fix $\mathtt{G}$ and let $\mathcal{W} \subseteq \mathtt{G}$. Let $\mathbb{E}_{\mathtt{E}'}[\cdot]$ denote the expectation with respect to the channel realizations $\mathtt{E}'$ in $\mathtt{G}\backslash\mathcal{W}$. Then

$$\mathbb{E}_{\mathtt{E}'}[M(\mathtt{G}, (\mathcal{W}, \mathtt{E}'), \mathcal{W}) \mathbb{1}_{\{\mathtt{E}' \in \mathcal{E}'_{\mathtt{G},\mathcal{W}}\}}] \leq \mathbb{E}_{\mathtt{E}'}[M(\mathtt{G}, (\mathcal{W}, \mathtt{E}'), \mathcal{W})] \mathbb{E}_{\mathtt{E}'}[\mathbb{1}_{\{\mathtt{E}' \in \mathcal{E}'_{\mathtt{G},\mathcal{W}}\}}]. \quad (5)$$

Discussion: The operational significance of this lemma is that in order to upper bound the size of the marking process we are free to consider the noise realization outside the witness to be independent of the witness.

G. Back to Expansion

Now where we have randomized the channel values we can use expansion arguments to deal with the dependence on the graph. The basic idea is simple. Assume that the neighborhood of initially bad edges (at the start of the marking process) is perfectly tree-like. This means that two bad edges never converge on the same variable node in their future. In this case the only bad messages emitted by a variable node are due to bad received values, but these received values can be thought of being chosen independently from the rest of the process. It follows that the whole marking process can be modeled as a birth and death process. When we grow out an edge then with probability ϵ we encounter a variable with a bad received value. In this case, the variable emits bad messages along its two outgoing edges and those in return each create $\mathtt{r} - 1$ bad outgoing messages at the output of their connected check nodes. In other words, with probability ϵ one bad edge is transformed to $2(\mathtt{r} - 1)$ bad edges. With probability $1-\epsilon$ the process along the particular edge dies. By the stability condition of the LGalB decoder $2(\mathtt{r}-1)\epsilon^{\text{LGalB}} \leq 1$. We conclude that the expected number of newly generated children is strictly less than 1 for $\epsilon < \epsilon^{\text{LGalB}}$. Therefore the corresponding birth and death process dies with probability 1.

Since in general the expansion of the local neighborhood is not perfectly tree-like, the above argument has to be extended to account for this. But the gist of the argument remains the same.

IV. Conclusion

We have shown two approaches for solving the problem of limit exchange below the DE threshold. The first one, based solely on the expansion property of the graph, helps in proving the result for a large class of MP decoders but only if the degree is relatively large. To prove the result for smaller degrees one has to include the role of channel realizations. The second approach accomplishes this in some cases. In this paper we only considered channel parameters below the DE threshold. But the regime above this threshold is equally interesting and important. One important application of proving the exchange of limits in this regime is the finite-length analysis via a scaling approach since the computation of the scaling parameters heavily depends on the fact that this exchange is permissible.

Acknowledgment

We would like to thank A. Montanari for interesting discussions. The work of S. Korada is supported by NCCR-MICS grant number 5005-67322.

References

[1] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2005, in preparation.

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

[3] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Inform. Theory*, vol. 8, pp. 21–28, jan 1962.

[4] R. J. McEliece, E. Rodemich, and J.-F. Cheng, "The turbo decision algorithm," in *Proc. of the Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, USA, 1995.

[5] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 782–790, Feb. 2001.