LDPC Decoders with Missing Connections

Linjia Chang, Avhishek Chatterjee, and Lav R. Varshney
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

***Abstract*—Due to process variation in nanoscale manufacturing, there may be permanently missing connections in information processing hardware. Due to timing errors in circuits, there may be missed messages in intra-chip communications, equivalent to transiently missing connections. In this work, we investigate the performance of message-passing LDPC decoders in the presence of missing connections. We prove concentration and convergence theorems that validate the use of density evolution performance analysis. Arbitrarily small error probability is not possible under miswiring, but we find suitably defined decoding thresholds for communication systems with binary erasure channels and peeling decoders, as well as binary symmetric channels and Gallager A decoders. We see that decoding is robust to missing connections, as decoding thresholds degrade smoothly.**

I. Introduction

Low-density parity-check (LDPC) codes are prevalent due to their performance near the Shannon limit with message-passing decoders having efficient implementation [1]. With the end of CMOS scaling near, there is interest in nanoscale circuit implementations of decoders, but this introduces concerns that process variation in manufacturing may lead to interconnect patterns different than designed [2]–[4], especially under self-assembly [5], [6]. Yield on manufactured chips deemed perfectly operational is small, leading to rather expensive industrial waste [7], but changing the paradigm of circuit functionality from perfection to some small probability α of missing connections may eliminate much wastage. It is of interest to characterize chips with permanently missing connections so that suitable error tolerances may be determined.

Process variation in manufacturing also causes fluctuation in device geometries which might prevent them from meeting timing constraints [8]. Such timing errors lead to missed messages in intra-chip communications, equivalent to transiently missing connections. It is also of interest to characterize decoders with transiently missing connections.

Most work in fault-tolerant computing assumes the circuit is constructed correctly and is concerned only with faults in computational elements; as Elias says in classical work [9], "papers assume that the wiring diagram is correctly drawn and correctly followed in construction, but that computation proper is performed only by unreliable elements." The only work we are aware of in fault-tolerant computing theory that (briefly) discusses wiring errors is the monograph of Winograd and Cowan [10].

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

We had previously extended the method of density evolution to decoders with faults in the computational elements and showed that it is possible to communicate with arbitrarily small error probability with noisy Gaussian belief propagation [11]. Asymptotic characterizations were also determined for Gallager A [11] and Gallager B decoders with transient noise [12], [13], and both permanent and transient noise [14]. General belief propagation, not necessarily in decoding, has also been studied [15]. Rather than noise in computational elements, here we analyze the performance of message-passing decoders with missing wiring and show that decoding thresholds are robust, in the sense of degrading smoothly. This is true for both transiently and permanently missing connections in message-passing decoding circuits.

The celebrated results of Richardson and Urbanke [16] developed density evolution for analyzing message-passing decoders for LDPC codes that are correctly wired. Here we extend those results, so we can use the density evolution technique to characterize symbol error rate P_e, measuring the fraction of incorrectly decoded symbols at the end of message-passing decoding, even when the decoder has missing connections. Traditionally [16], there are thresholds for channel noise level ε below which P_e can be driven to 0 with increasing blocklength n. Unfortunately with missing connections in message-passing decoders, P_e cannot be driven to 0 in general. Thus, following [11], we let η be an upper bound to the final error probability that can be achieved by decoders with missing connections after many iterations ℓ and give thresholds to ε, below which $\lim_{\ell\to\infty} P_e^{(\ell)} < \eta$ under density evolution.

Sec. II gives mathematical models of LDPC decoders with both transiently and permanently missing connections and develops concentration and convergence theorems that provide validity to density evolution analysis of P_e. Secs. III and IV analyze the peeling decoder on the binary erasure channel (BEC) and the Gallager A decoder on binary symmetric channel (BSC) using density evolution, characterizing P_e with missing connections. Sec. V concludes the paper by outlining directions for further investigation.

II. Model and Performance Analysis Tools

A. Ensemble of LDPC Codes and Channel

In this work we are concerned with the standard LDPC code ensemble of $(d_{\mathtt{v}}, d_{\mathtt{c}})$-regular codes of length n, which can be defined by a bipartite Tanner graph with variable nodes of degree $d_{\mathtt{v}}$ and check nodes of degree $d_{\mathtt{c}}$. Later we also discuss irregular codes, where the degree distribution of variable and check nodes are denoted by functions $\lambda(x) = \sum_{d=2}^{\infty} \lambda_d x^{d-1}$ and $\rho(x) = \sum_{d=2}^{\infty} \rho_d x^{d-1}$. We consider this binary linear code ensemble as over the alphabet $\{\pm 1\}$. Although results in this section are general, for convenience, let us think of the communication channel as either BSC with output alphabet $\{\pm 1\}$ or BEC with output alphabet $\{\pm 1, \text{"?"}\}$.

B. Decoder and Missing Connections

The decoder operates by passing messages over the Tanner graph of the code. For notational convenience, let us restrict attention to decoders with messages in $\{\pm 1, \text{"?"}\}$, but again concentration and convergence results are general.

Now we introduce missing connections between the check-node and variable-node modules. Connections may go missing either permanently or transiently due to failures of intra-chip communication. For a given decoder circuit, permanent failure is modeled by removing each connection between variable and check nodes with probability α independently from others, before decoding starts. These connections are never active. On the contrary, with transiently missing connections, each connection is removed independently from others with probability α during each iteration of the decoding algorithm. Our conversations with circuit designers suggest that when an interconnect is broken in LDPC decoders, the measured voltage at this open-ended wire is neither low nor high in circuit signals; it is some undefined floating value that may vary within a range. So, whenever there is a missing connection between a variable node and a check node, we assume that an erasure symbol "?" is exchanged.

C. Restriction to All-One Codeword

Under certain symmetry conditions of the code, the communication channel, and the message-passing decoder, the probability of error is independent of the transmitted codeword.

C1. **Code Symmetry:** Code is a binary linear code.
C2. **Channel Symmetry:** Channel is a binary memoryless symmetric channel [17, Def. 4.3 and 4.8].
C3. **Check Node Symmetry:** If incoming messages of a check node are multiplied by $\{b_i \in \{\pm 1\}\}$, then the computed message is multiplied by $\prod_i b_i$.
C4. **Variable Node Symmetry:** If the sign of each incoming message is flipped, the sign of the computed message is also flipped.

Proposition 1: Under conditions C1–C4, in the presence of transiently or permanently missing connections, the probability of error of a message passing decoder is independent of the transmitted codeword.

Proof: It follows by mapping to 0 the erasure message "?", sent when a connection is missing. Thus the check-to-variable and variable-to-check messages are the messages computed at check node and variable node, respectively, multiplied by either 1 (connection exists) or 0 (missing connections). Thus, messages passed between check and variable nodes satisfy the respective symmetry conditions [17, Def. 4.82]. Hence, the result follows by invoking [17, Lem. 4.92]. ■

D. Concentration around Ensemble Average

We now show that the performance of LDPC codes decoded with missing-connection decoders stays close to the expected performance of the code ensemble for both transiently and permanently missing connections. The approach follows [16] and is based on constructing an exposure Martingale, obtaining bounded difference constants, and using Azuma's inequality. Fix the number of decoding iterations at some finite ℓ and let Z be the number of incorrect values held among all $d_{\mathtt{v}}n$ variable nodes at the end of ℓth iteration for a specific choice of code, channel noise, and a decoder with missing connections. Let $E[Z]$ denote the expectation of Z.

Theorem 1 (Concentration Around Expected Value): There exists a positive constant $\beta = \beta(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ such that for any $\varepsilon > 0$,

$$\Pr[|Z - E[Z]| > nd_{\mathtt{v}}\varepsilon/2] \leq 2e^{-\beta\varepsilon^2 n}.$$

Proof Sketch: Recall the Doob's Martingale construction from [16], and the bounded difference constants for exposing channel noise realizations and the realized code connections, together with Azuma's inequality. The main difference here is in the bounded differences due to the additional randomness from missing connections.

For permanently missing connections, one can think of the final connection graph being sampled from an ensemble of irregular random graphs with binomial degree distribution with average degrees $(1-\alpha)d_{\mathtt{c}}$ and $(1-\alpha)d_{\mathtt{v}}$, bounded by maximum degrees $d_{\mathtt{c}}$ and $d_{\mathtt{v}}$. Hence, the result follows from the result for correctly-wired irregular codes [16].

For transiently missing connections, the Martingale is constructed differently. Here instead of edges, for ℓ iterations, we sequentially expose the realization of edges at different iterations. Similar to [11] for transient noise, the Martingale difference is bounded using the maximum number of edges over which a message can propagate in ℓ iterations, using the computation graph method. ■

Note β will be smaller for transient than permanent miswiring. The theorem extends directly to irregular LDPC codes.

E. Convergence to the Cycle-Free Case

We now show that the average performance of an LDPC code ensemble converges to an associated cycle-free tree structure, unwrapping a computation tree as in [16].

For an edge whose connected neighborhood with depth 2ℓ is cycle-free, let q denote the expected number of incorrect values held along this edge at the end of ℓth decoding iteration. The expectation is taken over the choice of code, the messages received from the channel, and the realization of the decoder with missing wires. The theorems hold for both transiently and permanently missing connections.

Theorem 2 (Convergence to Cycle-Free Case): There exists a positive constant $\gamma = \gamma(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ such that for any $\varepsilon > 0$ and $n > \frac{2\gamma}{\varepsilon}$,

$$|E[Z] - nd_{\mathtt{v}}q| < nd_{\mathtt{v}}\varepsilon/2.$$

Proof: The proof is identical to [16, Theorem 2]. ■

Note that the basic idea of the proof is to show the probability of repeats in the computation tree goes to zero with increasing graph girth. This further implies that the density evolution equations we will obtain for transiently and permanently missing connections will be identical. In particular, in density evolution the state variable $x_{\ell+1}$ is computed based on the x_ℓ of nodes immediately below in the infinite tree. Each connection in the tree is encountered only once, and in case of permanent failure each connection is present with probability $1-\alpha$.

Theorem 3 (Concentration around Cycle-Free Case): There exists positive constants $\beta = \beta(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ and $\gamma = \gamma(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ such that for any $\varepsilon > 0$ and $n > \frac{2\gamma}{\varepsilon}$,

$$\Pr[|Z - nd_{\mathtt{v}}q| > nd_{\mathtt{v}}\varepsilon] \leq 2e^{-\beta n\varepsilon^2}.$$

Proof: Follows directly from Theorems 1 and 2. ∎

This tree-ensemble concentration result holds for all message-passing decoders with missing connections. In the sequel, we consider the special cases of peeling and Gallager A decoders.

III. Peeling Decoder for Binary Erasure Channel

Consider the peeling decoder for a BEC(ε), with alphabet $\{\pm 1, \text{“?”}\}$. The check node computation is a product of all messages ± 1 it receives from neighboring variable nodes if none is “?”, otherwise “?”. The variable node computation is to send any ± 1 symbol received either from the other check nodes or from the channel, otherwise send ?. When the connection between two nodes is missing, the message exchanged is equivalent to “?”, so peeling extends naturally to decoders with missing connections. Note that this decoder satisfies the symmetry conditions C1–C4, so we can use density evolution for the all-one codeword.

First we see that even in the non-asymptotic regime, performance degrades with missing connections. Noise or miswiring enhancement observed elsewhere does not hold [18], [19].

Lemma 1: For any finite LDPC code and finite number of decoding iterations, for both permanently and transiently missing connections, $P_e^{(\ell)}(g, \varepsilon, \alpha)$ increases monotonically with α for a given ε.

Proof follows by coupling arguments. We couple two missing-connection processes for different α to get a sample path dominance of connections. Then we note the fact that since no erroneous message is transmitted in peeling decoders with perfect connections, missing connections can only degrade the performance.

A similar coupling argument yields an ordering relationship with respect to channel erasure probability ε for a given α.

A. Density Evolution

Recall a peeling decoder allows alphabet $\{\pm 1, \text{“?”}\}$: it only outputs either a correct or erasure symbol. Consider a regular $(d_{\mathtt{v}}, d_{\mathtt{c}})$ LDPC code, BEC(ε) channel with parameter ε, and each wire that can be disconnected with the node independently with probability α in each decoding iteration. Let $x_0, x_1, \ldots, x_\ell$ denote the fraction of erasures existing in the code at each decoding iteration. The original received message from the channel is erased with probability ε, so

$$P_e{}^{(0)}(\varepsilon, \alpha) = x_0 = \varepsilon.$$

Let q_{in} and q_{out} be the probabilities a node receives/sends an erasure, respectively. At a variable node, the probability a given internal incident variable is erased is the probability both the external incident variable is erased and all other $d_{\mathtt{v}}-1$ nodes are either disconnected or connected but erased.

$$q_{out} = \varepsilon[\alpha + (1-\alpha)q_{in}]^{d_{\mathtt{v}}-1}$$

At a check node, the probability a given incident variable will not be erased is the probability that all $(d_{\mathtt{c}}-1)$ other internal incident variables are not erased or disconnected. So the probability that a message is erased is

$$q_{out} = 1 - [(1-q_{in})(1-\alpha)]^{d_{\mathtt{c}}-1}.$$

Let $f_{DE}(x_\ell, \varepsilon, \alpha) = x_{\ell+1}$ be the recursive update function for the fraction of errors between two consecutive iterations:

$$\varepsilon\left[\alpha + (1-\alpha)\left(1 - [(1-x_\ell)(1-\alpha)]^{d_{\mathtt{c}}-1}\right)\right]^{d_{\mathtt{v}}-1},$$

or for irregular codes:

$$\varepsilon\lambda\left(\alpha + (1-\alpha)\left(1 - \rho[(1-x_\ell)(1-\alpha)]\right)\right).$$

It is clear that $f_{DE}(\varepsilon, x, \alpha)$ is non-decreasing in each of its arguments, given the other two. Thus, a monotonicity result similar to [17, Lem. 3.54] holds here. This in turn implies a convergence result for x_ℓ similar to [17, Lem. 3.56]. For a given α and ε, x_ℓ converges to the nearest fixed point, $x = f_{DE}(\varepsilon, x, \alpha)$.

Lemma 2: For any irregular code ensemble $C^\infty(\lambda, \rho)$, there exists a $\delta > 0$, such that the probability of error $P_e^{(\infty)}$ satisfies $P_e^{(\infty)} - \delta > \varepsilon\lambda(1-(1-\alpha)\rho(1-\alpha)) > 0$ with probability 1.

Proof: Since x_ℓ is monotonic, if $x_0 \leq x_1$ then for any ℓ, $x_{\ell+1} \geq x_\ell \geq x_{\ell-1}$. Now, for $x_0 = 0$, by substituting this value in f_{DE},

$$x_1 = f_{DE}(0, \varepsilon, \alpha) = \varepsilon\lambda(1-(1-\alpha)\rho(1-\alpha)) > 0 = x_0.$$

This implies that $\lim_{\ell\to\infty} x_\ell \geq f_{DE}(0, \varepsilon, \alpha)$, for $x_0 = 0$. But, as x_ℓ converges to the fixed point nearest to x_0, this implies there is no fixed point in $(0, f_{DE}(0, \varepsilon, \alpha))$ for any $\alpha > 0$. ∎

B. Detailed Characterization

Now we use our density evolution equation to study final error probability of decoders with missing connections. Clearly for a peeling decoder, when $\varepsilon = 0$, the error probability stays at 0 regardless of the quality of the decoder. When $\alpha = 0$, we recover the traditional decoding threshold [16]. Thus we consider the system when $\varepsilon > 0$ and $\alpha > 0$. Since it is impossible to drive error probability to 0, we use a weaker notation called the η-threshold [11], where small value η limits the final decoding error probability P_e. This threshold is defined to be:

$$\varepsilon^*(\eta, \alpha) = \sup\{\varepsilon \in [0, 0.5] | \lim_{\ell\to\infty} P_e^{(\ell)} < \eta\}. \qquad (1)$$

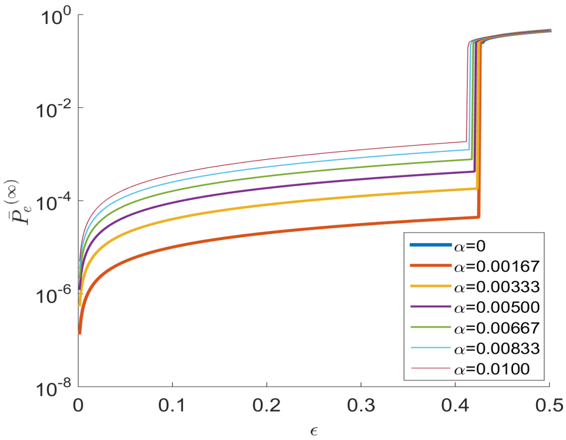

Fig. 1. Final symbol error rate of decoding a $C^{\infty}(3,6)$ LDPC code under peeling decoding algorithm with various missing connection probability α over BEC(ε).

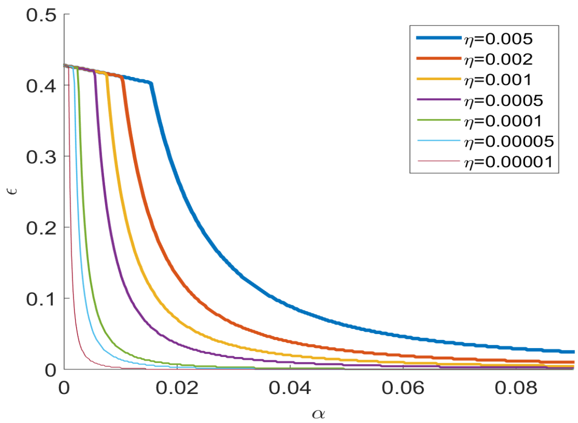

Fig. 2. Channel threshold of decoding a $C^{\infty}(3,6)$ LDPC code under peeling decoding algorithm over BEC(ε) for different given final error η-thresholds.

The fixed points can be found by solving for the real solutions to the following polynomial equation:

$$x - \varepsilon \left[\alpha + (1-\alpha)\left(1 - [(1-x)(1-\alpha)]^{d_c - 1}\right)\right]^{d_v - 1} = 0.$$

Figs. 1 and 2 demonstrate a threshold phenomenon and show that decoding is robust to missing connections. Results can also be used to cast a performance equivalence between resources to improve channel noise and decoder fidelity.

IV. Gallager A for Binary Symmetric Channel

Consider a modification of the Gallager A decoder for a BSC(ε), using an extended alphabet $\{\pm 1, \text{“?”}\}$. For correctly wired decoders, the check node computation is the XOR of incoming variable-to-check messages, to enforce the parity constraints of the code. When one of the symbols in the parity is an unknown “?”, that parity check is no longer informative since any bit of a binary linear code is equally likely to be ± 1. So, for decoders with missing connections we make a natural adaptation: a message from check node c to variable node v is “?” if any of the incoming messages from neighboring variable nodes of c are “?”, otherwise it is the usual XOR. For variable-to-check messages, a variable node v sends to c its received bit y_{v} from the channel if all other check node messages are not $-y_{\mathsf{v}}$, otherwise it sends $-y_{\mathsf{v}}$. We make a natural adaptation: send y_{v} if all of the other unerased check node messages are not $-y_{\mathsf{v}}$, otherwise send $-y_{\mathsf{v}}$. This modification of the Gallager A algorithm satisfies symmetry conditions C1–C4.

Unlike in the peeling decoder for BEC, messages between Gallager A nodes may be erroneous. So, for a sample path realization of channel and missing connections, it may happen that a missing connection prevents propagation of erroneous messages. Hence, it is not apparent there is a stochastic dominance result like Lem. 1 between two different probabilities of missing connections.

A. Density Evolution

Assuming the all-one codeword is sent, we find $x_{\ell+1}$, the probability for a variable node to compute -1 at iteration $(\ell+1)$, in terms of x_ℓ. First, note that a variable node in the Gallager A adaptation never computes “?”, even though it may receive (due to missing connections or check-node computation “?”) or send “?” (only due to missing connections). The probability of a check node computation being -1 is the probability all $(d_{\mathsf{c}}-1)$ variable nodes are connected and send odd number of -1, denoted by p_{-1}:

$$p_{-1} = (1-\alpha)^{d_{\mathsf{c}}-1}\frac{(1-(1-2x_\ell)^{d_{\mathsf{c}}-1})}{2},$$

Similarly, the probability of a check node computation being $+1$ is the probability all $(d_{\mathsf{c}}-1)$ variable nodes are connected and send even number of -1, denoted by p_{+1}:

$$p_{+1} = (1-\alpha)^{d_{\mathsf{c}}-1}\frac{(1+(1-2x_\ell)^{d_{\mathsf{c}}-1})}{2},$$

where the results follow using [1, Sec. 4.3].

The probability of a check-to-variable message being “?” is denoted by $p_0 = 1 - p_{+1} - p_{-1} = 1-(1-\alpha)^{d_{\mathsf{c}}-1}$.

Also, consider a binomial random variable $V \sim \mathcal{B}(d_{\mathsf{v}}-1, 1-\alpha)$ with probability density function $p_V(v)$ capturing the number of check nodes connected to a variable node.

Now at iteration $(\ell+1)$, the fraction of incorrect values held at a variable node is the sum of the probability of two events. The first event is that the message received from the channel is correct while none of the incoming messages from the connected check nodes is correct, but not all of them are “?” or only one says different while others are “?”, to break the tie when there is degree-one node. The second event is that the message received from the channel is wrong while at least *two* of the incoming messages from the connected check nodes are wrong, or all of them are “?”.

The probability of the first event is:

$$\begin{aligned}
&\mathbb{E}_V\Big[(1-\varepsilon)[\Pr\{\text{no connected check nodes sends } 1\} \\
&\quad - \Pr\{\text{all } V \text{ connected check nodes send “?”}\} \\
&\quad - \Pr\{\text{one check node sends } -1 \text{ while others send “?”}\}]\Big] \\
&= \sum_{v=1}^{d_v - 1} p_V(v)(1-\varepsilon)[(p_{-1}+p_0)^v - p_0^v - p_{-1}{p_0}^{v-1}].
\end{aligned}$$

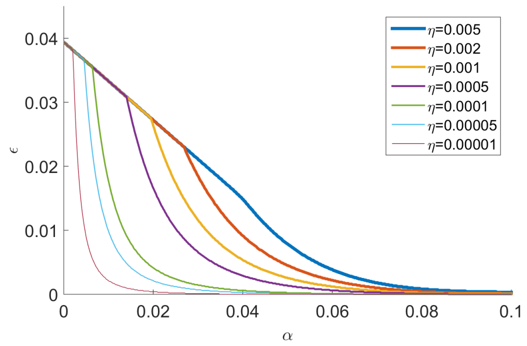

Fig. 3. η-thresholds for decoding a $C^\infty(3,6)$ regular LDPC code with α-missing wire Gallager A decoding algorithm over BSC(ε).

The probability of the second event is:

$$\mathbb{E}_V\left[\varepsilon[\Pr\{\text{at least one connected check nodes sends } -1\} + \Pr\{\text{all } V \text{ connected check nodes send “?”}\}]\right] = \sum_{v=0}^{d_v-1} p_V(v)\varepsilon[1-(p_{+1}+p_0)^v + p_0^v].$$

Let $x_{\ell+1} = f_{DE}(x_\ell, \varepsilon, \alpha)$, and take the expectation of V to get:

$$f_{DE} = \varepsilon\alpha^{d_v-1} + \sum_{v=1}^{d_v-1}\binom{d_v-1}{v}(1-\alpha)^v\alpha^{(d_v-1-v)}\Big[(1-\varepsilon)\left[(p_{-1}+p_0)^v - p_0^v - p_{-1}{p_0}^{v-1}\right] + \varepsilon[1-(p_{+1}+p_0)^v + p_0^v]\Big].$$

It can be seen that f_{DE} is monotonic in x for a given set of α and ε. Hence, by the same arguments as for peeling decoders, for any initial $0 \le \varepsilon = x_0 \le 0.5$, x_ℓ converges to the nearest fixed point of the density evolution equation. Also, note form the update equation that for $\varepsilon, \alpha > 0, x_\ell = 0$, $f(x_\ell, \varepsilon, \alpha) > 0$. This implies a result similar to Lem. 2 here.

B. Detailed Characterization

We carried out analysis to find η-thresholds for communication under the Gallager A decoding with missing connections. The result is in Fig. 3; recall for a $(3,6)$ regular LDPC code with a perfect Gallager A decoder, the threshold is roughly 0.039 [16]. Note that P_e can be driven to a small number even with the presence of missing wires. Decoding is robust to missing connections, though less than the peeling decoder over BEC.

V. Conclusion

This work analyzes message-passing decoders with both transient and permanent missing connections in hardware, deriving density evolution equations to characterize error probability in peeling decoders for BEC and Gallager A decoders for BSC. Although P_e cannot be driven to 0 in the presence of missing connections, it can be suppressed to a small value. That is, η-reliable communication is possible with faulty decoders with missing connections. In a sense, even when the encoder and decoder speak different languages, the result is not catastrophic.

Future work involves considering not just decoders with missing connections, but also miswired and noisy decoders. One may also design new decoder architectures to ensure reliable communication even with miswiring; for example, horizontal connections, a crucial structure in the cortex contributing to the filling in of the missing parts in visual images [20, Ch. 8.33], can be added to decoder designs.

Acknowledgment

Thanks to J. Cowan, A. C. Singer, and N. R. Shanbhag for encouragement.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] A. A. Al-Yamani, S. Ramsundar, and D. K. Pradhan, “A defect tolerance scheme for nanotechnology circuits,” *IEEE Trans. Circuits Syst. I*, vol. 54, no. 11, pp. 2402–2409, Nov. 2007.

[3] S.-L. Jeng, J.-C. Lu, and K. Wang, “A review of reliability research on nanotechnology,” *IEEE Trans. Rel.*, vol. 56, no. 3, pp. 401–410, Sep. 2007.

[4] Y. M. Chee and A. C. H. Ling, “Limit on the addressability of fault-tolerant nanowire decoders,” *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 60–68, Jan. 2009.

[5] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, “A defect-tolerant computer architecture: Opportunities for nanotechnology,” *Science*, vol. 280, no. 5370, pp. 1716–1721, Jun. 1998.

[6] M. Haselman and S. Hauck, “The future of integrated circuits: A survey of nanoelectronics,” *Proc. IEEE*, vol. 98, no. 1, pp. 11–38, Jan. 2010.

[7] M. A. Breuer, S. K. Gupta, and T. M. Mak, “Defect and error tolerance in the presence of massive numbers of defects,” *IEEE Des. Test. Comput.*, vol. 21, no. 3, pp. 216–227, May-June 2004.

[8] S. Ghosh and K. Roy, “Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era,” *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[9] P. Elias, “Computation in the presence of noise,” *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 346–353, Oct. 1958.

[10] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. Cambridge, MA: MIT Press, 1963.

[11] L. R. Varshney, “Performance of LDPC codes under faulty iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[12] S. M. S. Tabatabaei Yazdi, H. Cho, and L. Dolecek, “Gallager B decoder on noisy hardware,” *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1660–1673, May 2013.

[13] S. M. S. Tabatabaei Yazdi, C.-H. Huang, and L. Dolecek, “Optimal design of a Gallager B noisy decoder for irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2052–2055, Dec. 2012.

[14] C.-H. Huang, Y. Li, and L. Dolecek, “Gallager B LDPC decoder with transient and permanent errors,” *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 15–28, Jan. 2014.

[15] ——, “Belief propagation algorithms on noisy hardware,” *IEEE Trans. Commun.*, vol. 63, no. 1, pp. 11–24, Jan. 2015.

[16] 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.

[17] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge: Cambridge University Press, 2008.

[18] M. J. Wainwright, “Estimating the “wrong” graphical model: Benefits in the computation-limited setting,” *J. Mach. Learn. Res.*, vol. 7, pp. 1829–1859, Dec. 2006.

[19] H. Chen, L. R. Varshney, and P. K. Varshney, “Noise-enhanced information systems,” *Proc. IEEE*, vol. 102, no. 10, pp. 1607–1621, Oct. 2014.

[20] P. Brodal, *The Central Nervous System*. New York, NY: Oxford University Press, 2016.