Multi-Edge Type LDPC Codes

Tom Richardson *Senior member, IEEE* Rüdiger Urbanke

Abstract

We introduce multi-edge type LDPC codes, a generalization of the concept of irregular LDPC codes that yields improvements in performance, range of applications, adaptability and error floor. The multi-edge type LDPC codes formalism includes various constructions that have been proposed in the literature including, among others, concatenated tree (CT) codes [1], (generalized irregular) repeat accumulate (RA) codes [2], [3], and codes in [4], [5]. We present results establishing the improved performance afforded by this more general framework. We also indicate how the analysis of LDPC codes presented in [6], [7] extends to the multi-edge type setting.

Index Terms

Low density parity check code, belief propagation, irregular LDPC, threshold

I. Introduction

In this paper we introduce multi-edge type LDPC codes, a generalization of regular and irregular LDPC codes. The framework gives rise to ensembles not possible in the irregular LDPC framework. Various new features can be added to LDPC designs and new constraints brought to bear. This framework has already been used to produce LDPC codes that perform better than standard irregular LDPC codes over standard channels such as the AWGN channel, especially for short block lengths, while requiring lower complexity. It has been used to adapt an LDPC design to a turbo equalizer receiver structure, producing significant gains [8]. The framework yields high performance codes of very low rates and high rate codes with low error floors.

Tom Richardson is with Flarion Technologies, Bedminster, NJ 07921, Email: tjr@flarion.com

Rüdiger Urbanke is with EPFL, LTHC-DSC, CH-1015 Lausanne, Email: Rudiger.Urbanke@epfl.ch

In the recent literature on LDPC codes one finds various constructions whose structure falls outside the scope of standard irregular LDPC as considered in, e.g., [7], [9]. Through these constructions, and also through observation of the structure of optimized degree distributions of standard irregular LDPC codes, it has become apparent that improvements in various aspects of the iterative decoding performance of LDPC codes can be achieved by introducing new features into the graph specifying the code. Some such recently proposed constructions are concatenated tree (CT) codes [1], (generalized irregular) repeat accumulate (RA) codes [2], [3], and certain codes constructed in [4], [5] that we will refer to as KS codes. The multi-edge type generalization admits all of the above-mentioned constructions as special cases. The multi-edge formalism is capable of specifying a particular graph; Thus, the formalism is in some sense universal. The purpose, however, is to specify ensembles, i.e., to design macroscopic interconnection structure.

The multi-edge formalism admits parametric optimization as done for standard irregular LDPC codes (degree distributions) in [7]. We use density evolution to find interesting top-performing structures and present various examples: structures with high thresholds using only low degrees (low complexity), structures that simultaneously yield high thresholds and low error floors, and structures useful for very low rate LDPC codes. The framework also provides for better adaptation of code structure to a larger iterative process, such as a turbo equalizer, for an example see [8].

The basic idea of multi-edge type LDPC codes is inherent in the modifier “multi-edge type.” In the definition of standard irregular LDPC ensembles (degree distributions) all edges in the Tanner graph representation are statistically interchangeable. In other words, there exists a single statistical equivalence class of edges. Here we consider ensembles with multiple equivalence classes of edges. In the standard irregular LDPC ensemble definition, nodes in the graph (both variable and constraint) are specified by their degree, i.e., the number of edges they are connected to. In the multi-edge type setting an *edge degree* is a vector; it specifies the number of edges connected to the node from each edge equivalence class (type) independently. A formal definition of the multi-edge type LDPC ensemble is presented in Section II.

Standard irregular LDPC code ensembles are specified by their *degree distributions*. These are polynomials representing the fraction of nodes of each degree, usually taken from the *edge perspective*. In the multi-edge type setting the edge perspective is problematic because there is no unique edge perspective and it often occurs that no single edge type connects to every node type. Therefore, in the multi-edge type setting we prefer to represent graph ensembles from a *node perspective*. We introduce our parameterization in Section II. Under this parameterization all structural constraints on the parameters are linear and key functionals, such as code rate, are linear. In standard irregular LDPC codes the degree distributions are constrained by the fact that, in an actual graph, the number of edges emanating from the variable node side must coincide with the number of edges emanating from the constraint node side. In the multi-edge type setting, each edge type introduces one such constraint.

Let us briefly remark on what we consider to be the main significance and advantage of the multi-edge type extension of LDPC ensembles. Optimized (for threshold) standard irregular LDPC code ensembles tend to have a large fraction of degree two variable nodes. Introducing a positive fraction of degree one variable nodes in the degree distribution of standard irregular LDPC ensembles guarantees that the threshold will cease to exist; it is impossible, under these circumstances, for the infinite graph to converge to zero probability of error. It is evident, however, in view of the predominance of degree two variable nodes in optimized degree sequences that degree one variable nodes might be beneficial. Degree one variable nodes can be admitted without sacrificing the threshold if they are admitted in a controlled way. The multi-edge type generalization of LDPC codes provides for this extension. Other structural modifications afforded by multiple edge types include the ‘accumulate’ structure associated to RA codes. This structure can be interpreted as connecting degree two variable nodes in a chain (this idea also appears in the constructions of KS codes [5]). This method of controlling degree two variable nodes can be shown to have benefit on the stability of the ‘delta-function-at-infinity’ fixed point of density evolution, allowing a relatively large number of degree two nodes without loosing stability, and it often simplifies encoding. On the other hand, large numbers of degree two variable nodes tend to produce graphs with relatively high error floors. This too can be addressed in the multi-edge type framework. A third modification multi-edge type LDPC codes readily admit is the addition punctured variable nodes, sometimes referred to as state variables. This will be seen to be useful both for increasing thresholds and for lowering error floors. Another generalization we admit is to allow different node types to have different received distributions associated to them. This is the mechanism by which we introduce punctured variable nodes. This can also be useful in, e.g., combining multi-level modulation with LDPC coding. We shall not, however, pursue this extension in this paper.

II. MULTI-EDGE TYPE LDPC ENSEMBLES

Bi-partite (Tanner) [10] graphs represent LDPC codes in a natural way, with variable nodes on the one hand (left), constraint nodes on the other (right), and edges between the two indicating participation of variables in constraints. Let us recall the structure of irregular LDPC graphs. The *degree* d of node specifies the number of edges it attaches to. We say that the node has d sockets: edges attach to sockets. The number of sockets K on either side of the graph is the same. An edge is a pairing of sockets, one from each side of the graph. If node degrees are specified then one often considers *ensembles* of graphs. If the sockets on both sides are enumerated then a specific graph is determined by the permutation on K elements giving the pairing of sockets and an ensemble is induced by a set of permutations.

A multi-edge type ensemble is comprised of a finite number $n_{\mathfrak{e}}$ of edge types. The *degree type* of a constraint node is vector of (non-negative) integers of length $n_{\mathfrak{e}}$; the ith entry of this vector records the number of sockets of the ith type connected to such a node. We shall also refer to this vector as an *edge degree* in this case. The *degree type* of a variable node has two parts although it may be viewed as a vector of (non-negative) integers of length $n_{\mathfrak{r}} + n_{\mathfrak{e}}$, where $n_{\mathfrak{r}}$ is the number of different channels over which a bit may be transmitted. The two parts are of length $n_{\mathfrak{r}}$ and $n_{\mathfrak{e}}$ respectively. The first part relates to the received distribution and will be termed the *received degree* and the second part specifies the *edge degree*. The edge degree plays the same role as for constraint nodes. Edges are typed as they pair sockets of the same type. This constraint, that sockets must pair with sockets of like type, characterizes the multi-edge type concept. In a multi-edge type description, different node types may have different received distributions, i.e, the associated bits may go through different channels.

We assume throughout that all channels are symmetric (see Section VII for a definition and [7] for further implications). They are therefore represented by R the distribution of log-likelihood ratio $z(y) := \log \frac{p(y|x=0)}{p(y|x=1)}$ conditioned on $x = 0$.

In this paper we focus on graphs with two received distributions. The first, whose density we denote by R_0, is special as it is used is used to indicate state (or punctured) variables. The channel associated to R_0 is erasure with probability one, hence, $\mathsf{R}_0 = \delta_0$. The second, whose density is R_1, represents the binary memoryless symmetric channel over which the code bits are transmitted.

III. Parameterization

To represent the structure of the graph we introduce the following node-perspective multinomial representation. We interpret degrees as exponents. Let $\mathbf{d} := (d_1, ..., d_{n_{\mathfrak{e}}})$ be a multi-edge degree and let $\mathbf{x} := (x_1, ..., x_{n_{\mathfrak{e}}})$ denote (vector) variables. We use $\mathbf{x}^{\mathbf{d}}$ to denote $\prod_{i=1}^{n_{\mathfrak{e}}} x_i^{d_i}$. Similarly, Let $\mathbf{b} := (b_0, ..., b_{n_{\mathfrak{r}}})$ be a received degree and let $\mathbf{r} := (r_0, ..., r_{n_{\mathfrak{r}}})$ denote variables corresponding to received distributions. By $\mathbf{r}^{\mathbf{b}}$ we mean $\prod_{i=0}^{n_{\mathfrak{r}}} r_i^{b_i}$. Typically, vectors $\mathbf{b}$ will have one entry set to 1 and the rest set to 0.

A graph ensemble is specified through two multinomials, one associated to variable nodes and the other associated to constraint nodes. We denote these multinomials by

$$\nu(\mathbf{r}, \mathbf{x}) := \sum \nu_{\mathbf{b},\mathbf{d}}\, \mathbf{r}^{\mathbf{b}} \mathbf{x}^{\mathbf{d}} \qquad \text{and} \qquad \mu(\mathbf{x}) := \sum \mu_{\mathbf{d}}\, \mathbf{x}^{\mathbf{d}}$$

respectively, where coefficients $\nu_{\mathbf{b},\mathbf{d}}$ and $\mu_{\mathbf{d}}$, are non-negative reals.

We will now interpret the coefficients. Let N be the length of the codeword. For each variable node degree type $(\mathbf{b}, \mathbf{d})$ the quantity $\nu_{\mathbf{b},\mathbf{d}} N$ is the number of variable nodes nodes of type $(\mathbf{b}, \mathbf{d})$ in the graph. Similarly, for each constraint node degree type $\mathbf{d}$ the quantity $\mu_{\mathbf{d}} N$ is the number of constraint nodes of type $\mathbf{d}$ in the graph.

Let us introduce some additional notation. We define

$$\nu_{r_i}(\mathbf{r}, \mathbf{x}) := \frac{d}{dr_i}\nu(\mathbf{r}, \mathbf{x})\,, \;\; \nu_{x_i}(\mathbf{r}, \mathbf{x}) := \frac{d}{dx_i}\nu(\mathbf{r}, \mathbf{x})\,, \quad \text{and } \mu_{x_i}(\mathbf{x}) := \frac{d}{dx_i}\mu(\mathbf{x})\,.$$

We use $\mathbf{1}$ to denote a vector of all 1's, the length being determined by context.

The coefficients of ν and μ are constrained to ensure that the number of sockets of *each type* is the same on both sides (variable and constraint) of the graph. This gives rise to $n_{\mathfrak{e}}$ linear conditions on the coefficients of ν and μ as follows.

[Socket count equality constraints] $$\nu_{x_i}(\mathbf{1}, \mathbf{1}) = \mu_{x_i}(\mathbf{1}), \qquad i = 1, ..., n_{\mathfrak{e}}\,. \tag{1}$$

When there are multiple (non-empty) channels, with the ith being used for a fraction π_i of the transmitted bits, then we have the constraints

[Received Constraints] $$\nu_{r_i}(\mathbf{1}, \mathbf{1}) = \pi_i\,, \;\; i = 1, ..., n_{\mathfrak{r}}\,, \tag{2}$$

where, typically, π_i will be given positive constants satisfying $\sum_{i=1}^{n_{\mathfrak{r}}} \pi_i = 1$. When there is only a single channel then the constraint reduces to $\nu_{r_1}(\mathbf{1}, \mathbf{1}) = 1$. Note that ν_{r_0} is not directly constrained, we are free to introduce punctured variable nodes.

The nominal code rate (assuming all constraints are linearly independent) is given by

$$\text{[Code Rate]} \qquad \texttt{rate} = \nu(\mathbf{1},\mathbf{1}) - \mu(\mathbf{1})\,. \tag{3}$$

To see that this is the code rate, note that if we multiply $\texttt{rate}$ by N then the result is the number of variable nodes in the graph minus the number of constraint nodes. This is, nominally, the number of independent variable nodes. Dividing this by N is, by definition, the code rate. Thus, for a given set of node degree types, the set of possible code rates is an interval whose minimum and maximum may be found by linear programming. Fixing the code rate amounts to imposing a linear constraint on the parameters.

Note that all key functionals and constraints on the parameters are linear. Thus, even after fixing the code rate, the space of admissible parameters is a convex polytope. Extreme points may be easily found using linear programming. This aspect of the parameterization is particularly useful when optimizing the parameters.

The specific multi-edge type structures we present in this paper have relatively few node degree types. For this reason, when presenting structures we list only those degrees that have non-zero coefficients in ν and μ. Consider the example shown in Table VI. This example has four edge types. Thus, the edge degree vectors, presented in the rows of the table, are non-negative integer sequences of length four.

IV. Graph Ensembles

A variable node of degree type $(\mathbf{b},\mathbf{d})$ has d_i sockets of type i. A constraint node of degree type $\mathbf{d}$ has d_i sockets of type i. The parameters are constrained so that, for each edge type, the total number of sockets on the variable node side is the same as the total number of sockets on the constraint node side.

Consider enumerating the sockets on both sides of the graph in some arbitrary manner. Let the total number of sockets on either side of the graph be $K = K_1 + K_2 + ... + K_{n_\mathfrak{e}}$, where K_i is the number of edges of type i. A particular graph is determined by a permutation π on K letters such that socket i on the variable node side is connected to socket $\pi(i)$ on the constraint node side. In the multi-edge type setting we constrain socket $\pi(i)$ to be of the same type as socket i. Thus, $\pi = (\pi_1, ..., \pi_{n_\mathfrak{e}})$ can be decomposed into $n_\mathfrak{e}$ distinct permutations where π_i is a permutation on K_i elements.

The standard ensemble of graphs under consideration is defined by viewing π_i as a random variable distributed uniformly over the set of all permutations on K_i elements. Furthermore, the permutations for different edge types are independent. In practice, of course, one does not choose the permutations π_i uniformly randomly.

V. Concentration

The analysis carried out for standard LDPC ensembles in [6] and [7] extends straightforwardly to multi-edge type LDPC codes. The *concentration theorem* for LDPC codes holds for the generalized structures. Some details of the proof presented in [6], [7] require modification: a different phase in the martingale argument is required for each edge type.

Suppose we fix the parameters ν, μ describing multi-edge type structure and we instantiate a particular graph of size $n.$ Let $Z_\ell[n]$ denote the bit error rate after decoding ℓ iterations. We may interpret $Z_\ell[n]$ as averaged over the realization of the channel or not; only the constants in the statement of the theorem below will depend on this choice. Let $\mathbb{E}Z_\ell[n]$ denote the expected value of this quantity, averaged over the choice of permutation $\pi,$ i.e., over the ensemble of graphs (assuming fixed nodes), and also over the realization of the channel.

Theorem 1: For any ℓ there exists a constant β such that the probability that $\big|Z_\ell[n]-\mathbb{E}[Z_\ell[n]]\big| > \epsilon$ is less than $e^{-\beta\epsilon^2 n}$. Furthermore, the limit $Z_\ell[\infty] := \lim_{n\to\infty} \mathbb{E}[Z_\ell[n]]$ exists and there exists a constant γ such that $\big|\mathbb{E}[Z_\ell[n]] - Z_\ell[\infty]\big| < \frac{\gamma}{n}$.

VI. Distributions

The primary objects of interest in the analysis of decoding LDPC codes are probability distributions of log-likelihood ratios, and symmetric distributions (see the next section) in particular. The space of these distributions and its basic properties were presented in [7]. We recall the basic definitions here.

Let $\mathcal{F}$ denote the space of right continuous, non-decreasing functions F defined on $\mathbb{R}$ satisfying $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) \leq 1$. To each $F \in \mathcal{F}$ we associate a random variable z over $(-\infty, +\infty]$. The random variable z has *law* or *distribution* $F,$ i.e., Pr$\{z \in (-\infty, x]\} = F(x).$ The reason we allow $\lim_{x\to\infty} F(x) \leq 1$ rather than $\lim_{x\to\infty} F(x) = 1$ is to permit z to have some probability mass at $+\infty,$ indeed Pr$\{z = +\infty\} = 1 - \lim_{x\to\infty} F(x).$ A random variable z over $(-\infty, +\infty]$ is completely specified by its distribution $F_z \in \mathcal{F}.$

We will work with "densities" over $(-\infty, +\infty]$ which, formally, can be treated as (Radon-Nikodyn) derivatives of elements of $\mathcal{F}$. Certain densities appear often in the analysis. The density δ_0 represents the derivative of the distribution $F = 1_{x \geq 0}$ and corresponds to a channel that erases with probability 1. The density δ_∞ represents the derivative of the distribution $F = 0$ and corresponds to a channel that delivers bits perfectly.

We say that a sequence of densities $f_1, f_2, \ldots$ converges to the limit f if $F_1(x), F_2(x), \ldots$ converges to $F(x)$ at all points of continuity of F.

The main properties of $\mathcal{F}$ we use in this paper are sequential compactness, that any sequence of densities has a convergent subsequence, and monotonicity under physical degradation, that any sequence of densities ordered by physical degradation converges to a limit density.

VII. Symmetry

We assume BPSK signaling and that all channels are symmetric, that is

$$p(y|x=1) = p(-y|x=-1)\,.$$

Without loss of generality, we assume that y is the log-likelihood ratio of x, and denote $p(y|x = 1)$ by $\mathsf{R}(y)$. It follows that R is a symmetric density, i.e.,

$$\mathsf{R}(-y) = e^{-y}\mathsf{R}(y)\,.$$

Since all densities and channels considered in this paper are symmetric, we will use symmetric densities to represent channels and vice-versa.

Let P be a symmetric density. Define $\Pr_{\text{err}}(\mathsf{P})$ to be the propability of error of the associated channel,

$$\Pr_{\text{err}}(\mathsf{P}) := \int_{-\infty}^{0-} \mathsf{P}(x)\,dx \,+\, \frac{1}{2}\int_{0-}^{0+} \mathsf{P}(x)\,dx\,. \tag{4}$$

Let us also define the *Battacharya constant* associated to the channel:

$$\mathcal{B}(\mathsf{P}) := \int_{-\infty}^{+\infty} e^{-\frac{1}{2}x}\mathsf{P}(x)\,dx\,. \tag{5}$$

For symmetric distributions the following bound holds.

$$2\Pr_{\text{err}}(\mathsf{P}) \leq \mathcal{B}(\mathsf{P}) \leq 2\sqrt{\Pr_{\text{err}}(\mathsf{P})(1-\Pr_{\text{err}}(\mathsf{P}))} \tag{6}$$

where the left inequality is tight for the binary erasure channel (BEC) and the right is tight for the binary symmetric channel (BSC). (For a proof see [7].)

Let us consider the performance of a repetition code over the channel P. If we repeat a bit k times independently over the channel P (canonically assuming the bit is a 0 and transmitted using BPSK signalling, i.e., as a 1) then its posterior likelihood has distribution $\mathsf{P}^{\otimes k}$, P convolved with itself (over $\mathbb{R}$) k times. We have the following result

Lemma 1: If Q corresponds to any symmetric channel, then

$$\mathrm{Pr}_{\mathrm{err}}(\mathsf{Q}^{\otimes \ell}) \geq \mathcal{B}(\mathsf{Q})^{\ell} \left(\frac{1}{1 + \frac{e^2 \mathcal{B}(\mathsf{Q})}{4(\ell+1)}} \right) \left(\frac{e}{3\pi} \sqrt{\frac{\mathcal{B}(\mathsf{Q})}{\ell+1}} \right)$$

This result is useful in the analysis of the stability condition of density evolution, see Section IX. The proof of this result is given in the appendix.

The following result was proved in [7]

Lemma 2: If Q is a symmetric channel with probability of error ϵ, then Q is physically degraded with respect to $\eta\delta_0 + (1-\eta)\delta_\infty$ for all $\eta \leq 2\epsilon$.

A. Analyticity of Symmetric Densities

Let f be a symmetric density. The generalized Fourier transform of f,

$$\hat{\mathsf{f}}(z) := \int_{-\infty}^{\infty} e^{-zt} \mathsf{f}(t)\, dt\,,$$

is analytic in the strip $\{z : \Re z \in (0,1)\}$. Note that, by symmetry, this function is symmetric about the point $z = (\frac{1}{2}, 0)$. Note also that $\mathcal{B}(\mathsf{f}) = \hat{\mathsf{f}}(\frac{1}{2}, 0)$. Let us define $\mathcal{B}^c(\mathsf{f}) := \hat{\mathsf{f}}(\frac{1}{2} + c, 0)$ for $c \in (-\frac{1}{2}, \frac{1}{2})$. Thus, $\mathcal{B}^0(\mathsf{f}) = \mathcal{B}(\mathsf{f})$ and $\mathcal{B}^c(\mathsf{f}) = \mathcal{B}^{-c}(\mathsf{f})$.

Theorem 2: A symmetric density f is uniquely determined by $\mathcal{B}^c(\mathsf{f})$ for any interval of values for c.

Proof: The density f is determined by $\hat{\mathsf{f}}(z)$ for $\{z : \Re z \in (0,1)\}$, and, since this analytic, it is determined by $\mathcal{B}^c(\mathsf{f})$ for any infinite set of c with an accumulation point inside $(0,1)$. ■

Theorem 3: The functionals $\mathcal{B}^c(\mathsf{f})$ are monotonic under physical degradation, i.e., if $\mathsf{f} \succ \mathsf{g}$, then $\mathcal{B}^c(\mathsf{f}) \geq \mathcal{B}^c(\mathsf{g})$ for all c.

Proof: The proof will be presented elsewhere. ■

VIII. Density Evolution

Density evolution [6] describes the behavior of the decoder in the infinite block length limit. Density evolution therefore allows determination of $Z_\ell[\infty]$. (We assume here and henceforth that a belief propagation decoder is used.) An efficient algorithm implementing density evolution for standard LDPC ensembles was described in [6] and further refined in [7]. An even more efficient implementation has been recently described in [12].

Under density evolution, the distributions, or message densities, propagate through the abstract description of the graph in a manner paralleling the propagation of messages through the graph under message passing decoding. The main difference between the two is that message densities emerging from the different node types are averaged: In standard irregular LDPC codes they are averaged across the different degrees; In multi-edge type LDPC they are averaged across node types independently for each edge type. In the standard irregular LDPC framework messages are represented by a single density function, since all edges are statistically equivalent; In the multi-edge type framework messages are represented by $n_\mathfrak{e}$ density functions, one for each edge type.

We shall here briefly describe how message densities are combined at the nodes. Under belief propagation, incoming log-likelihood ratio messages at variable nodes are summed to give outgoing log-likelihood ratio messages, hence, under density evolution, the message densities are convolved. Under belief propagation, at constraint nodes we can transform the log-likelihoods messages into a representation over $\mathrm{GF}[2]\times\mathbb{R}^+$ so that the message passing operation performed on the messages is again addition. Thus, under density evolution, the densities of the messages, when view as densities over $\mathrm{GF}[2]\times\mathbb{R}^+$, are again convolved. (For details see [7].) To distinguish these two convolutions we refer to them as convolution in the variable node domain, in the first case, and convolution in the constraint node domain, in the second case.

Let us introduce some additional notation. We consider vectors of symmetric densities $\mathsf{\mathbf{P}} = (\mathsf{P}_1, ..., \mathsf{P}_{n_\mathfrak{e}})$ where P_i is the density of messages carried on edge type i. By $\boldsymbol{\delta}_0$ we mean a vector of densities where each density is δ_0. Similarly, by $\boldsymbol{\delta}_\infty$ we mean a vector of densities where each density is δ_∞.

We introduce the following multinomials:

$$\begin{aligned}\boldsymbol{\lambda}(\mathbf{r},\mathbf{x}) &:= \left(\frac{\nu_{x_1}(\mathbf{r},\mathbf{x})}{\nu_{x_1}(\mathbf{1},\mathbf{1})}, \frac{\nu_{x_2}(\mathbf{r},\mathbf{x})}{\nu_{x_2}(\mathbf{1},\mathbf{1})}, ..., \frac{\nu_{x_{n_\mathfrak{e}}}(\mathbf{r},\mathbf{x})}{\nu_{x_{n_\mathfrak{e}}}(\mathbf{1},\mathbf{1})}\right) \\ \boldsymbol{\rho}(\mathbf{x}) &:= \left(\frac{\mu_{x_1}(\mathbf{x})}{\mu_{x_1}(\mathbf{1})}, \frac{\mu_{x_2}(\mathbf{x})}{\mu_{x_2}(\mathbf{1})}, ..., \frac{\mu_{x_{n_\mathfrak{e}}}(\mathbf{x})}{\mu_{x_{n_\mathfrak{e}}}(\mathbf{1})}\right) \ .\end{aligned} \tag{7}$$

We remark that in the case of a single edge type, these definitions reduce to degree distribution pairs (λ, ρ) of standard irregular LDPC codes except that we have introduced a variable representing received distributions into the arguments of λ. Previously, the received distribution was not an argument of λ.

Let $\mathsf{P}^{\ell}(\mathsf{Q})$ denote the vector of messages passed from variable nodes to check nodes in iteration ℓ assuming that $\mathsf{P}^{0}(\mathsf{Q}) = \mathsf{Q}$. When no argument is indicated we mean $\mathsf{Q} = \boldsymbol{\delta}_0$, corresponding to an actual decoding. Similarly, let R denote distributions corresponding to the received distributions. $\big(\mathsf{R}(y) = p(y|x=1)$ assuming that $y = \log \frac{p(y|x=1)}{p(y|x=-1)}\big)$.

The following recursion represents density evolution.

$$\text{[Density Evolution]} \qquad \mathsf{P}^{\ell+1} = \boldsymbol{\lambda}(\mathsf{R}, \boldsymbol{\rho}(\mathsf{P}^{\ell}))\,. \tag{8}$$

Here, the multiplication operation inside $\boldsymbol{\rho}(\mathsf{P})$ is convolution of densities in the constraint node domain and inside $\boldsymbol{\lambda}(\mathsf{R}, \mathsf{X})$ it is convolution of densities in the variable node domain.

Density evolution describes the asymptotic behavior of the decoding process for the multi-edge type ensemble as block length tends to infinity as described in [6].

A. Fixed Points, Convergence, and Monotonicity

In the multi-edge type setting certain degeneracies arise that are not possible in the standard irregular LDPC setting. There are two obvious degeneracies we will rule out a priori: Punctured degree one variable nodes and received distributions equal to δ_∞. A punctured degree one variable node effectively eliminates its neighboring constraint from the decoding process, and a node with received distribution δ_∞ can be removed from the graph with no effect.

We say Q is *physically degraded* with respect to P, denoted $\mathsf{Q} \succ \mathsf{P}$, if Q_i is physically degraded with respect to P_i, denoted $\mathsf{Q}_i \succ \mathsf{P}_i$, for $i = 1, ..., n_{\mathfrak{e}}$. We assume a parameterized family of input distributions $\mathsf{R}(\sigma)$ ordered by physical degradation where increasing σ degrades the input distribution.

Theorem 4: If, for a vector density Q and some $k > 0$ we have $\mathsf{P}^{k}(\mathsf{Q}) \prec (\succ)\mathsf{Q}$, then, $\mathsf{P}^{k(\ell+1)}(\mathsf{Q}) \prec (\succ)\mathsf{P}^{k\ell}(\mathsf{Q})$ for all $\ell \geq 0$, and $\mathsf{P}^{\ell}(\mathsf{Q}) \xrightarrow{\ell\to\infty} \mathsf{F}$ for some vector density F.

Proof: Monotonicty, that $\mathsf{P}^{k(\ell+1)}(\mathsf{Q}) \prec (\succ)\mathsf{P}^{k\ell}(\mathsf{Q})$ under the stated assumptions, follows from the tree channel argument appearing in [11]. Convergence of $\mathsf{P}^{k\ell}(\mathsf{Q})$ in ℓ follows from sequential compactness of the space of symmetric distributions and completeness of functionals monotonic under physical degradation. Finally, convergence of P^{ℓ} follows from continuity of density evolution. ■

The above theorem implies that $\mathsf{P}^{\ell}(\boldsymbol{\delta}_0)$ always converges to a well-defined limit density. Usually we are interested in knowing when the output bit error rate goes to zero, i.e., we are interested in knowing when $\nu(\mathsf{R}, \boldsymbol{\rho}(\mathsf{P}^{\ell})) \xrightarrow{\ell \to \infty} \delta_\infty$.

Assume $\mathsf{R} = \mathsf{R}(\sigma)$ is a one parameter family of vector distributions ordered by physical degradation: $\mathsf{R}(\sigma) \succ \mathsf{R}(\sigma')$ if $\sigma > \sigma'$. We define the decoding *threshold* as

$$\sigma^* := \sup\{\sigma : \lim_{\ell \to \infty} \nu(\mathsf{R}(\sigma), \boldsymbol{\rho}(\mathsf{P}^{\ell})) = \delta_\infty\}.$$

In the standard irregular LDPC setting, if $\sigma < \sigma^*$ then $\mathsf{P}^{\ell} \xrightarrow{\ell \to \infty} \delta_\infty$. In the multi-edge setting it is not always obvious what the fixed-point limit of P^{ℓ} might be. It is clear, however, that such a fixed point must be *perfectly decodable.*

A density vector F is a *perfectly decodable* fixed point of density evolution if it is a fixed point and $\nu(\mathbf{R}, \boldsymbol{\rho}(\mathsf{F})) = \delta_\infty$. A multi-edge type structure may have more than one perfectly decodable fixed point. Indeed, consider the simple example

$$\nu(r_0, r_1, x_1, x_2) = r_0 x_1 x_2, \qquad \text{and} \qquad \mu(x_1, x_2) = x_1 x_2.$$

Any vector density $\mathsf{F} = (\mathsf{F}_1, \mathsf{F}_2)$ satisfying $\mathsf{F}_1 = \delta_\infty$ or $\mathsf{F}_2 = \delta_\infty$ is a perfectly decodable fixed point for the associated density evolution.

Given a fixed point F, let

$$\mathcal{E}_\infty^{\text{VC}} = \mathcal{E}_\infty^{\text{VC}}(\mathsf{F}) := \{i : F_i = \delta_\infty\} \qquad \text{and} \qquad \mathcal{E}_{<\infty}^{\text{VC}} = \mathcal{E}_{<\infty}^{\text{VC}}(\mathsf{F}) := \{i : F_i \neq \delta_\infty\}\,. \tag{9}$$

Similarly, let

$$\mathcal{E}_\infty^{\text{CV}} = \mathcal{E}_\infty^{\text{CV}}(\mathsf{F}) := \{i : \rho_i(\mathsf{F}) = \delta_\infty\} \qquad \text{and} \qquad \mathcal{E}_{<\infty}^{\text{CV}} = \mathcal{E}_{<\infty}^{\text{CV}}(\mathsf{F}) := \{i : \rho_i(\mathsf{F}) \neq \delta_\infty\}\,. \tag{10}$$

Lemma 3: If the fixed point F is perfectly decodable then

$$\mathcal{E}_{<\infty}^{\text{VC}}(\mathsf{F}) \cap \mathcal{E}_{<\infty}^{\text{CV}}(\mathsf{F}) = \emptyset.$$

Proof: Assume F is perfectly decodable. Since no received distribution is δ_∞, we see that each variable node type has positive degree in $\mathcal{E}_\infty^{\text{CV}}$. It follows that if $i \notin \mathcal{E}_\infty^{\text{CV}}$ then $i \in \mathcal{E}_\infty^{\text{VC}}$, i.e., $\mathcal{E}_{<\infty}^{\text{VC}} \subset \mathcal{E}_\infty^{\text{CV}}$. ■

Note that a corollary of the above lemma is that no constraint node type can have degree more than one in $\mathcal{E}_{<\infty}^{\text{VC}}(\mathsf{F})$ for a perfectly decodable fixed point F.

One of the complicating features of the multi-edge framework is the possibility of having degree one variable nodes. The presence of degree one variable nodes guarantees that message distributions associated with certain edge types will be strictly bounded away from δ_∞. Any edge type that is connected to degree one variable nodes never carries the distribution δ_∞ in the variable-to-check direction. Any other edge type that connects to constraint nodes also connected to these edge types never carries the distribution δ_∞ in the check-to-variable direction. This "bounded away from δ_∞" propagates through the graph in a manner similar to erasure decoding with check nodes and variable nodes reversed.

Recursively define two edge-type subsets $\mathcal{E}^{\text{VC}}_{\text{fin}}$ and $\mathcal{E}^{\text{CV}}_{\text{fin}}$ as follows. Initially $\mathcal{E}^{\text{VC}}_{\text{fin}}[0]$ is all edge types that have positive probability of being connected to a degree one variable node and $\mathcal{E}^{\text{CV}}_{\text{fin}}[0] = \emptyset$.

Iterate the following two steps for $t = 1, 2, \ldots$

1) Let $\mathcal{E}^{\text{CV}}_{\text{fin}}[t]$ be the union of $\mathcal{E}^{\text{CV}}_{\text{fin}}[t-1]$ and those edge types $i \notin \mathcal{E}^{\text{CV}}_{\text{fin}}[t-1]$ that have positive probability of being connected to a constraint node that has another edge in $\mathcal{E}^{\text{VC}}_{\text{fin}}[t-1]$.
2) Let $\mathcal{E}^{\text{VC}}_{\text{fin}}[t]$ be the union of $\mathcal{E}^{\text{VC}}_{\text{fin}}[t-1]$ and those edge types $i \notin \mathcal{E}^{\text{VC}}_{\text{fin}}[t-1]$ that have positive probability of connecting to a variable node having all other edges in $\mathcal{E}^{\text{CV}}_{\text{fin}}[t]$.

For all t large enough we have $\mathcal{E}^{\text{VC}}_{\text{fin}}[t] = \mathcal{E}^{\text{VC}}_{\text{fin}}[t-1]$ and $\mathcal{E}^{\text{CV}}_{\text{fin}}[t] = \mathcal{E}^{\text{CV}}_{\text{fin}}[t-1]$ and we let $\mathcal{E}^{\text{VC}}_{\text{fin}}$ and $\mathcal{E}^{\text{CV}}_{\text{fin}}$ denote these limiting sets.

Lemma 4: There exists a perfectly decodable fixed point if and only if $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$. Moreover, if $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$, then there exists a unique perfectly decodable fixed point $\mathbf{F}$ satisfying $\mathsf{F}_i = \delta_\infty$ for all $i \notin \mathcal{E}^{\text{VC}}_{\text{fin}}$.

Proof: It is clear that, for any fixed point $\mathbf{F}$, we have $\mathcal{E}^{\text{VC}}_{\text{fin}} \subset \mathcal{E}^{\text{VC}}_{<\infty}$ and $\mathcal{E}^{\text{CV}}_{\text{fin}} \subset \mathcal{E}^{\text{CV}}_{<\infty}$. Thus, Lemma 3 implies that there cannot exist a perfectly decodable fixed point if $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} \neq \emptyset$.

Assume now that $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$. It follows that no constraint node can have degree more than one in $\mathcal{E}^{\text{VC}}_{\text{fin}}$. It also follows that each variable node type has at least one edge type not in $\mathcal{E}^{\text{CV}}_{\text{fin}}$.

Let Ψ denote the space of vector densities $\mathbf{Q}$ satisfying $\mathsf{Q}_i = \delta_\infty$ for $i \notin \mathcal{E}^{\text{VC}}_{\text{fin}}$. For each $\mathbf{Q} \in \Psi$ we have $\rho_j(\mathbf{Q}) = \delta_\infty$ for all $j \notin \mathcal{E}^{\text{CV}}_{\text{fin}}$ and this implies that $\lambda_i(\mathbf{R}, \boldsymbol{\rho}(\mathbf{Q})) = \delta_\infty$ for $i \notin \mathcal{E}^{\text{VC}}_{\text{fin}}$. Thus, we see that $\mathbf{Q} \in \Psi \Rightarrow \mathbf{P}^1(\mathbf{Q}) \in \Psi$. Moreover, since every variable node type has at least one edge not in $\mathcal{E}^{\text{CV}}_{\text{fin}}$, we have $\nu(\mathbf{R}, \boldsymbol{\rho}(\mathbf{Q})) = \delta_\infty$ for all $\mathbf{Q} \in \Psi$.

Let $\mathbf{Q}^0 \in \Psi$ be that element with $\mathsf{Q}_i = \delta_0$ for $i \in \mathcal{E}^{\text{VC}}_{\text{fin}}$ and let $\mathbf{Q}^\infty \in \Psi$ be that element with $\mathsf{Q}_i = \delta_\infty$ for $i \in \mathcal{E}^{\text{VC}}_{\text{fin}}$. By Theorem 4 $\mathbf{P}^\ell(\mathbf{Q}^0) \xrightarrow{\ell\to\infty} \mathbf{F}^0$ and $\mathbf{P}^\ell(\mathbf{Q}^\infty) \xrightarrow{\ell\to\infty} \mathbf{F}^\infty$ for vector densities $\mathbf{F}^0, \mathbf{F}^\infty \in \Psi$ and any other fixed point $\mathbf{F} \in \Psi$ satisfies $\mathbf{F}^\infty \prec \mathbf{F} \prec \mathbf{F}^0$. What remains is to prove uniqueness, which is equivalent to $\mathbf{F}^\infty = \mathbf{F}^0$.

Since no check node can have degree more than one in $\mathcal{E}^{\text{VC}}_{\text{fin}}$, it follows that for any $\mathbf{Q} \in \Psi$ each $\rho_k(\mathbf{Q})$ is a convex combination of δ_∞ and $\mathsf{Q}_j, j \in \mathcal{E}^{\text{VC}}_{\text{fin}}$. Hence, for each $i \in \mathcal{E}^{\text{VC}}_{\text{fin}}$, $\mathsf{P}^1_i(\mathbf{Q})$ can be written as a polynomial in $\mathbf{R}$ and $\mathsf{Q}_j, j \in \mathcal{E}^{\text{VC}}_{\text{fin}}$.

$$\mathsf{P}^1_i(\mathbf{Q}) = \sum_{k=1}^{K_i} w_{i,k} \mathbf{R}^{\mathbf{c}^{i,k}} \mathbf{Q}^{\mathbf{e}^{i,k}}$$

where $w_{i,k} > 0$ and $\sum_{k=1}^{K_i} w_{i,k} = 1$ and multiplication denotes variable node domain convolution. Moreover, we can assume that $\mathbf{e}^{i,k}_j > 0$ only for $j \in \mathcal{E}^{\text{VC}}_{\text{fin}}$.

By induction, the same holds for $\mathsf{P}^\ell_i(\mathbf{Q})$ for each $\ell \geq 1$. We claim that for some ℓ sufficiently large $\mathsf{P}^\ell_i(\mathbf{Q})$ has at least one non-trivial term that does not depend on $\mathbf{Q}$, i.e., $w_{i,k} > 0$ and $\mathbf{e}^{i,k} = 0$. To see the claim first note that it is clearly true for each $i \in \mathcal{E}^{\text{VC}}_{\text{fin}}[0]$ for all $\ell \geq 1$. For each $i \in \mathcal{E}^{\text{CV}}_{\text{fin}}[1]$ and $\ell \geq 1$ it follows that $\rho^\ell_i(\mathbf{P}(\mathbf{Q}))$ is a polynomal in $\mathbf{R}$ and $\mathbf{Q}$ with at least one term that does not depend on $\mathbf{Q}$. It now follows that the claim holds for for each $i \in \mathcal{E}^{\text{VC}}_{\text{fin}}[1] \backslash \mathcal{E}^{\text{VC}}_{\text{fin}}[0]$ for all $\ell \geq 2$. Finite induction, following this line, completes the proof of the claim.

Let $\tilde{\mathbf{Q}}$ denote the sub-vector of $\mathbf{Q}$ taking only those components in $\mathcal{E}^{\text{VC}}_{\text{fin}}$. Thus, any perfectly decodable fixed point $\mathbf{F}$ we have

$$\tilde{\mathbf{F}} = \mathbf{U}(\tilde{\mathbf{F}}) \tag{11}$$

where

$$\mathsf{U}_i(\tilde{\mathbf{F}}) = w_{i,0} A_i(\mathbf{R}) + \sum_{k=1}^{K_i} w_{i,k} \mathbf{R}^{\mathbf{c}^{i,k}} \tilde{\mathbf{F}}^{\mathbf{e}^{i,k}} \tag{12}$$

where $A_i(\mathbf{R})$ is a fixed distribution, not δ_0, $w_{i,0} > 0$ and and $\sum_{k=0}^{K_i} w_{i,k} = 1$. The claim is that such an equation has a unique solution. Let $\mathbf{U}^\ell(\tilde{\mathbf{Q}})$ denote the ℓth iterate of $\mathbf{U}$, where $\mathbf{U}^0(\tilde{\mathbf{Q}}) = \tilde{\mathbf{Q}}$. Then $\mathbf{U}^\ell(\boldsymbol{\delta}_0)$ is a physically increasing sequence in ℓ that converges to the fixed point $\tilde{\mathbf{F}}^0$ and $\mathbf{U}^\ell(\boldsymbol{\delta}_\infty)$ is a physically decreasing sequence in ℓ that converges to the fixed point $\tilde{\mathbf{F}}^\infty$. Clearly, by monotonicity, for any fixed point $\tilde{\mathbf{F}}$ we have $\tilde{\mathbf{F}}^\infty \prec \tilde{\mathbf{F}} \prec \tilde{\mathbf{F}}^0$. We will show that $\tilde{\mathbf{F}}^\infty = \tilde{\mathbf{F}}^0$ by showing that $\mathcal{B}^c(\tilde{\mathbf{F}}^\infty) = \mathcal{B}^c(\tilde{\mathbf{F}}^0)$ for all $c \in (0,1)$.

Fix c. Let us abuse notation slightly by defining $\mathbf{U}(x)$ for a real valued vector x where multiplication in (12) is multiplication of reals as follows

$$\mathsf{U}_i(x) = w_{i,0}\mathcal{B}^c(A_i(\mathbf{R})) + \sum_{k=1}^{K_i} w_{i,k}\mathcal{B}^c(\mathbf{R})^{\mathbf{c}^{i,k}} x^{\mathbf{e}^{i,k}} . \tag{13}$$

First note for any c that $\mathcal{B}^c(\tilde{\mathbf{F}}^0)$ and $\mathcal{B}^c(\tilde{\mathbf{F}}^\infty)$ are fixed points of $\mathbf{U}$. Let $\mathbf{b}^0$ denote $\mathcal{B}^c(\tilde{\mathbf{F}}^0)$ and $\mathbf{b}^\infty$ denote $\mathcal{B}^c(\tilde{\mathbf{F}}^\infty)$. Note that if x and y are vectors satisfying $\mathbf{0} \le x \le y \le \mathbf{1}$ component-wise, then $\mathbf{0} < \mathbf{U}(x) \le \mathbf{U}(y) < \mathbf{1}$. Thus, for any fixed point $\mathbf{b}$ we have

$$\mathbf{0} < \mathbf{b}^\infty \le \mathbf{b} \le \mathbf{b}^0 < \mathbf{1} \tag{14}$$

Our aim is now to prove that $\mathbf{b}^0 = \mathbf{b}^\infty$. Note that U_i is an increasing convex function of its (real) arguments for each i. Since $\mathsf{U}(\mathbf{b}^0) = \mathbf{b}^0$ and $\mathsf{U}(\mathbf{b}^\infty) = \mathbf{b}^\infty$ it then follows that for each that $\mathbf{1} \ge \mathsf{U}(\mathbf{b}_0 + \epsilon(\mathbf{b}_0 - \mathbf{b}^\infty)) \ge (\mathbf{b}_0 + \epsilon(\mathbf{b}_0 - \mathbf{b}^\infty))$ component-wise for all for $\epsilon > 0$ sufficiently small But this implies that $\mathbf{U}^\ell(\mathbf{b}^0 + \epsilon(\mathbf{b}^0 - \mathbf{b}^\infty))$ converges to a fixed point $\mathbf{b}^*$ satisfying $\mathbf{b}^* \ge \mathbf{b}^0$ component-wise with equality holding only if $\mathbf{b}^\infty = \mathbf{b}^0$. By (14), however, we have $\mathbf{b}^* = \mathbf{b}^0$. ■

We summarize the general situation in the following.

Theorem 5: Given a multi-edge type structure, there exists a perfectly decodable fixed point if and only if $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$. Given $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$, the perfectly decodable fixed point is unique or there exist non-empty disjoint edge types $\mathcal{E}_1$ and $\mathcal{E}_2$, both disjoint from $\mathcal{E}^{\text{VC}}_{\text{fin}} \cup \mathcal{E}^{\text{CV}}_{\text{fin}}$, such that

A. No node type has degree more than one in $\mathcal{E}_1$.

B. All variable nodes with degree one in $\mathcal{E}_1$ are state variable nodes with all other edges in $\mathcal{E}_2$.

C. All check nodes with positive degree in $\mathcal{E}_2$ have degree one in $\mathcal{E}_1$.

Furthermore, if edge types satisfying A,B, and C, exist, then either there are multiple perfectly decodable fixed points or no perfectly decodable fixed point is reachable under standard (decoding) density evolution.

Proof: The first part of the theorem just restates Lemma 4.

Assume $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$. By Lemma 4 there exists a unique perfectly decodable fixed point $\mathbf{F}^\star$ with $\mathsf{F}^\star_i = \delta_\infty$ for all $i \notin \mathcal{E}^{\text{VC}}_{\text{fin}}$.

Assume now that there exists another perfectly decodable fixed point F and let G denote $\boldsymbol{\rho}(\mathsf{F})$. Then, $\mathsf{F}_k \neq \delta_\infty$ for some $k \notin \mathcal{E}_{\text{fin}}^{\text{VC}}$. Set

$$m := \max_{i \notin \mathcal{E}_{\text{fin}}^{\text{VC}}} \mathcal{B}(\mathsf{F}_i)\,.$$

and define

$$\mathcal{E}_1 := \{i \notin \mathcal{E}_{\text{fin}}^{\text{VC}} : \mathcal{B}(\mathsf{F}_i) = m\}\,.$$

Note that $\mathcal{E}_1$ is necessarily disjoint from $\mathcal{E}_{\text{fin}}^{\text{CV}}$ by Lemma 3. Now, an edge type $i \notin \mathcal{E}_{\text{fin}}^{\text{VC}}$ connects only to variable node types having at least one other edge not in $\mathcal{E}_{\text{fin}}^{\text{CV}}$. Therefore,

$$m \leq \max_{i \notin \mathcal{E}_{\text{fin}}^{\text{CV}}} \mathcal{B}(\mathsf{G}_i)\,. \tag{15}$$

An edge type $i \notin \mathcal{E}_{\text{fin}}^{\text{CV}}$ can not connect to constraint nodes having positive degree in $\mathcal{E}_{\text{fin}}^{\text{VC}}$, and, by Lemma 3, no constraint node type can have degree more than one in $\mathcal{E}_{<\infty}^{\text{VC}}(\mathsf{F})$. Therefore,

$$\max_{i \notin \mathcal{E}_{\text{fin}}^{\text{CV}}} \mathcal{B}(\mathsf{G}_i)\; \leq m\,. \tag{16}$$

Combining (15) and (16) we now have

$$\max_{i \notin \mathcal{E}_{\text{fin}}^{\text{CV}}} \mathcal{B}(\mathsf{G}_i) = m$$

and so we define

$$\mathcal{E}_2 := \{i \notin \mathcal{E}_{\text{fin}}^{\text{CV}} : \mathcal{B}(\mathsf{G}_i) = m\}\,.$$

By Lemma 3 we have $\mathcal{E}_2 \cap \mathcal{E}_{\text{fin}}^{\text{VC}} = \emptyset$.

We will now show that A, B, and C hold. First, since $\mathcal{E}_1 \subset \mathcal{E}_{<\infty}^{\text{VC}}(\mathsf{F})$, Lemma 3 implies no constraint node type can have degree more than one in $\mathcal{E}_1$. Let $(\mathbf{b},\mathbf{d})$ be a variable node type that has an edge of type $i \in \mathcal{E}_1$. Then variable node type $(\mathbf{b},\mathbf{d})$ has at least one other edge type not in $\mathcal{E}_{\text{fin}}^{\text{CV}}$. Thus, the density S outgoing from variable node type $(\mathbf{b},\mathbf{d})$ on the given edge in $\mathcal{E}_1$ at the fixed point F satisfies $\mathcal{B}(\mathsf{S}) \leq m$ by equation (16). Now, if $\mathcal{B}(\mathsf{S}) < m$ in any such case then we would have $\mathcal{B}(\mathsf{F}_i) < m$ which contradicts $i \in \mathcal{E}_1$. Since $\mathcal{E}_1 \subset \mathcal{E}_{\infty}^{\text{VC}}$, however, if a variable node type $(\mathbf{b},\mathbf{d})$ had degree more than one in $\mathcal{E}_1$, then, for variable node type $(\mathbf{b},\mathbf{d})$, we would have $\mathcal{B}(\mathsf{S}) = 0$. Since this cannot occur, no variable node type can have degree more than one in $\mathcal{E}_1$. We have now shown A.

By the same argument as above, we see that if a variable node type $(\mathbf{b},\mathbf{d})$ with an edge in $\mathcal{E}_1$, has a received distribution other than R_0 then $\mathcal{B}(\mathsf{S}) < m$. Since variable node types $(\mathbf{b},\mathbf{d})$

with an edge in $\mathcal{E}_1$ have at least one other edge in $\mathcal{E}_2,$ if they have a third edge type whose constraint-to-variable message density is not $\delta_0,$ then again we would have $\mathcal{B}(\mathsf{S}) < m.$ Now, only edge types in $\mathcal{E}_2$ can have constraint-to-variable message density equal to $\delta_0,$ so B holds. Note, moreover, that if $m < 1$ then each variable node type $(\mathbf{b}, \mathbf{d})$ with positive degree in $\mathcal{E}_1$ must have edge degree two.

Since any check node has degree at most one in $\mathcal{E}^{\text{VC}}_{<\infty},$ and edge types not in $\mathcal{E}^{\text{CV}}_{\text{fin}}$ connect only to constraint nodes with no edges in $\mathcal{E}^{\text{VC}}_{\text{fin}},$ we see that constraint nodes with positive degree in $\mathcal{E}_2$ must have precisely degree one in $\mathcal{E}_1,$ i.e., C holds.

Assume now that edge types satisfying A,B, and C exist. It is easy to see that $\mathsf{P}^{\ell}(\boldsymbol{\delta}_0)$ satisfies $\mathsf{P}^{\ell}_i = \delta_0$ for all $\ell \geq 0$ and $i \in \mathcal{E}_1.$ Thus, there is either a perfectly decodable fixed point F with $\mathsf{F}_i = \delta_0$ for all $i \in \mathcal{E}_1,$ or, if not, then $\mathsf{P}^{\ell}(\boldsymbol{\delta}_0)$ cannot converge to a perfectly decodable fixed point. ■

We will call a multi-edge structure *non-degenerate* if $\mathcal{E}^{\text{VC}}_{\text{fin}} \cap \mathcal{E}^{\text{CV}}_{\text{fin}} = \emptyset$ and there do not exists non-empty disjoint edge types $\mathcal{E}_1$ and $\mathcal{E}_2,$ both disjoint from $\mathcal{E}^{\text{VC}}_{\text{fin}} \cup \mathcal{E}^{\text{CV}}_{\text{fin}}$ satisfying A,B and C, of Theorem 5. Otherwise, the structure is called *degenerate.*

B. Cascaded Constructions for Degree One Variable Nodes

In most cases in which a graph is designed to have degree one variable nodes a simple cascaded construction is used. We now focus on this special case.

Let us assume that the Tanner graph can be decomposed into disjoint subgraphs Ξ_1 and Ξ_2 and the set of edges E_{12} which connect to both subgraphs. We say that the Tanner graph is a (Ξ_1, Ξ_2) cascade if all edges in E_{12} have their variable socket in Ξ_1 and their constraint socket in $\Xi_2.$

Typcially, we assume that the edge types E_{12} which connect to both to Ξ_1 and Ξ_2 only so connect, i.e., they are distinct from edge types connecting within Ξ_1 or $\Xi_2.$ We will be especially interested in the case where Ξ_2 contains variable nodes of degree one only, constraint nodes in Ξ_2 each connect to a single degree one variable node, and Ξ_1 contains variable nodes only of degree greater than one. Thus, the LDPC code consists of that portion defined by the graph $\Xi_1,$ together some parity check bits of the bits associated to $\Xi_1.$ We call such a construction a simple degree one cascade.

Obviously, the cascade concept can be recursed and generalized in other ways, but the construction outlined above is sufficient for the purposes of this paper and all of the examples presented later having degree one variable nodes are simple cascades.

IX. STABILITY

Stability analysis of density evolution examines the (asymptotic) behavior of the decoder when it is close to successful decoding. The basic question of interest is when can one be certain that the error probability is converging to zero, assuming it has gotten sufficiently close. In the standard irregular setting one can look at the error rate of the edge message distribution and ask whether it is tending to zero. In the multi-edge type setting this notion is inadequate because, as we saw in the previous section, messages on certain edge types may not converge to zero error probability even though the output error rate converges to zero. Nevertheless, a more-or-less complete stability theory can be developed. We shall present the theory for the case of simple cascades.

Consider first a multi-edge structure with no degree one variable nodes. Define the matrix $\Lambda = \Lambda(\mathbf{r})$ by

$$\Lambda_{i,j} := \frac{d}{dx_j}\lambda_i(\mathbf{r},\mathbf{x}) \mid_{\mathbf{x}=0}$$

and define the (constant) matrix $\mathcal{P}$ by

$$\mathcal{P}_{i,j} := \frac{d}{dx_j}\rho_i(\mathbf{x}) \mid_{\mathbf{x}=\mathbf{1}}$$

Theorem 6: Assume a multi-edge structure with no degree one variable nodes. If $\Lambda(\mathcal{B}(\mathsf{R}))\,\mathcal{P}$ is a stable matrix (spectral radius less than one) then there exists $\gamma > 0$ such that $\mathsf{P}_\ell(\mathsf{Q}) \xrightarrow{\ell\to\infty} \boldsymbol{\delta}_\infty$ for any Q satisfying $\max_i\{\mathrm{Pr}_{\mathrm{err}}(\mathsf{Q}_i)\} < \gamma$.

Proof: The proof of this result follows closely the proof for standard irregular LDPC ensembles which was presented in [11]. We will therefore not present the proof here. ■

Remark. Note that if the structure is degenerate, possessing edge type subsets $\mathcal{E}_1$ and $\mathcal{E}_2$ as in Theorem 5, then the matrix product $\Lambda(\mathcal{B}()R)\,\mathcal{P}$ is, at best, marginally stable, i.e., it has 0 as an eigenvalue.

The converse to Theorem 6 is a little trickier to state. It is not difficult to concoct structures where setting the density on one edge type to δ_∞ causes all densities to eventually converge to δ_∞ after a few iterations. Thus, assuming that $\Lambda(\mathcal{B}(\mathsf{R})\,\mathcal{P}$ is unstable, we cannot say, as we did in the standard irregular case, that if $\mathbf{Q} \neq \boldsymbol{\delta}_\infty$ then $\mathbf{Q}$ will not approach $\boldsymbol{\delta}_\infty$. We do, however, have the following.

Theorem 7: Assume a non-degenerate multi-edge structure with no degree one variable nodes. If $\Lambda(\mathcal{B}(\mathbf{R}))\,\mathcal{P}$ is an unstable matrix (spectral radius greater than one), then there exists $\epsilon > 0$ such that

$$\liminf_{\ell\to\infty} \text{Pr}_{\text{err}}(\nu(\mathbf{P}^{\ell}(\mathbf{Q}))) \geq \epsilon,$$

for any $\mathbf{Q}$ satisfying $\min_i\{\text{Pr}_{\text{err}}(\mathsf{Q}_i)\} > 0$.

Proof: Assume that $\Lambda(\mathcal{B}(\mathbf{R}))\mathcal{P}$ is unstable. By the Perron-Frobenious theorem there exists a non-negative (real) eigenvector $\mathbf{v}$ of $\Lambda(\mathcal{B}(\mathbf{R}))\mathcal{P}$ with real positive eigenvalue $\xi > 1$ whose magnitude equals the spectral radius. Without loss of generality we assume that $\mathbf{v}_i \leq 1$ for $i = 1, ..., n_{\mathfrak{e}}$.

Consider density vector $\mathbf{V}(\eta)$ where $\mathsf{V}_i(\eta) = \eta\mathbf{v}_i\delta_0 + (1-\eta\mathbf{v}_i)\delta_\infty$ defined for any $\eta \in [0,1]$. For any fixed ℓ, it is not hard to see that $\mathbf{P}^{\ell}(\mathbf{V}(\eta)) = \eta\big(\Lambda(\mathbf{R})\mathcal{P}\big)^{\ell}\mathbf{v} + O(\eta^2)$ where multiplication of matrix elements is variable domain convolution. It follows directly that $\mathcal{B}(\mathbf{P}^{\ell}(\mathbf{V}(\eta))) = \eta\xi^{\ell}\mathbf{v} + O(\eta^2)$. Analogous to Lemma 1, one can show (for a proof see the Appendix) that

$$\text{Pr}_{\text{err}}(\big(\Lambda(\mathbf{R})\mathcal{P}\big)^{\ell}\mathbf{v}) \geq \frac{1}{1+\frac{e^2\mathcal{B}_{\min}}{4(\ell+n_{\mathfrak{r}})}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}_{\min}}{\ell+n_{\mathfrak{r}}}}\,\xi^{\ell}\mathbf{v}$$

where the inequality is component-wise and $\mathcal{B}_{\min} := \min_i \mathcal{B}(\mathsf{R}_i)$. Since $\xi > 1$, we can choose $\ell = L$ large enough so that $\text{Pr}_{\text{err}}(\mathbf{P}^{L}(\mathbf{V}(\eta))) \geq 2\eta\mathbf{v} + O(\eta^2)$ component-wise. Assuming now that $\eta < \eta'$, where $\eta' > 0$ is small enough, we have $\text{Pr}_{\text{err}}(\mathbf{P}^{L}(\mathbf{V}(\eta))) \geq \eta\mathbf{v}$. It now follows from Lemma 2 that $\mathbf{P}^{L}(\mathbf{V}(\eta)) \succ \mathbf{V}(2\eta)$. By Theorem 4 we have $\mathbf{P}^{\ell}(\mathbf{V}) \xrightarrow{\ell\to\infty} \mathbf{F}$ for some fixed point $\mathbf{F}$ satisfying $\mathbf{F} \succ \mathbf{V}(\eta')$. Since $\mathbf{F} \neq \boldsymbol{\delta}_\infty$, $\mathbf{F}$ is not perfectly decodable. Let $\epsilon := \text{Pr}_{\text{err}}(\nu(\rho(\mathbf{F}(\eta'))))$ and note $\epsilon > 0$.

Assume $\min_i\{\text{Pr}_{\text{err}}(\mathsf{Q}_i)\} > 0$. By Lemma 2 we have $\mathbf{Q} \succ \mathbf{V}(\eta)$ for some $\eta \in (0, \eta']$. For ℓ sufficiently large we have $\mathbf{P}^{\ell}(\mathbf{V}(\eta)) \succ \mathbf{V}(\eta')$. Thus,

$$\begin{aligned}\liminf_{\ell\to\infty} \text{Pr}_{\text{err}}(\nu(\mathbf{P}^{\ell}(\mathbf{Q}))) &\geq \liminf_{\ell\to\infty} \text{Pr}_{\text{err}}(\nu(\mathbf{P}^{\ell}(\mathbf{V}(\eta)))) \\ &\geq \liminf_{\ell\to\infty} \text{Pr}_{\text{err}}(\nu(\mathbf{P}^{\ell}(\mathbf{V}(\eta')))) \\ &= \epsilon\,.\end{aligned}$$

■

A. Stability of Simple Cascades

In this section we study stability of (non-degenerate) simple cascaded constructions as described in Section VIII-B.

Consider a simple cascaded construction in which edge types $1, \ldots, j_1$ are those exclusively in Ξ_1, edge types $j_1+1, \ldots, j_2$ are those exclusively in $E_{1,2}$, and edge types $j_2, \ldots, n_{\mathfrak{e}}$ are those exclusively in Ξ_2, i.e., they attach exclusively to degree one variable nodes. Under density evolution, the densities $\mathsf{P}_i^{\ell}(\boldsymbol{\mathsf{Q}})$ for $i > j_2$ are independent of ℓ and $\boldsymbol{\mathsf{Q}}$, and each is a convex combination of the received densities $\mathsf{R}_1, \ldots, \mathsf{R}_{n_{\mathfrak{r}}}$. Without loss of generality (split edge types if necessary) we may assume that for $i > j_2$ and $\ell \geq 1$, $\mathsf{P}_i^{\ell}(\boldsymbol{\mathsf{Q}})$ a received density.

Let $\Psi(\boldsymbol{\mathsf{R}})$ denote the space of vector densities $\boldsymbol{\mathsf{Q}}$ of length $n_{\mathfrak{e}}$ such that $\mathsf{Q}_i = \mathsf{R}_{(i)}$ for $i \in [j_2+1, ..., n_{\mathfrak{e}}]$.

Given a vector $\mathbf{x} = (x_1, \ldots, x_K)$ we let $x_{[a,b]}$ denote the sub-vector $(x_a, \ldots, x_b)$.

The polynomial ν can be written as

$$\nu(\mathbf{r}, \mathbf{x}) = \nu^1(\mathbf{r}, x_{[1,j_2]}) + \nu^2(\mathbf{r}, x_{[j_2+1,n_{\mathfrak{e}}]})\,,$$

where each term in ν^1 has degree at least two in $x_{[1,j_1]}$ and each term in ν^2 has degree one in $x_{[j_2+1,n_{\mathfrak{e}}]}$.

Similarly, the polynomial μ can be written as

$$\mu(\mathbf{x}) = \mu^1(x_{[1,j_1]}) + \mu^2(x_{[j_1+1,n_{\mathfrak{e}}]})$$

and each term in μ^2 has degree one in $x_{[j_2+1,n_{\mathfrak{e}}]}$.

Let $\boldsymbol{\mathsf{F}}$ denote the unique perfectly decodable fixed point and let $\boldsymbol{\mathsf{G}} = \boldsymbol{\rho}(\boldsymbol{\mathsf{F}})$. Thus, $\mathsf{F}_i = \delta_\infty$ for $i \in 1, \ldots, j_2$, and for $i > j_2$ we have F_i is a received distribution. Define $\boldsymbol{\rho}^{2,1}(\mathbf{r}, \mathbf{x}_{[j_1+1,\ldots,j_2]}) := \left(\rho_{[j_1+1,j_2]}(\mathbf{x})\right)$. Let us now define the j_1 edge type, $n_{\mathfrak{r}}$ received type, multi-edge structure

$$\nu^s(\mathbf{r}, \mathbf{x}'), \qquad \mu^s(\mathbf{x}')$$

by

$$\begin{aligned} \nu^s(\mathbf{r}, \mathbf{x}') &:= \frac{1}{\nu^1(\mathbf{1}, \mathbf{1})} \nu^1(\mathbf{r}, \mathbf{x}', \boldsymbol{\rho}^{2,1}(\mathbf{r}, \mathbf{1})) \\ \mu^s(\mathbf{x}') &:= \frac{1}{\mu^1(\mathbf{1})} \mu^1(\mathbf{x}')\,. \end{aligned} \tag{17}$$

Essentially, this system represents the structure Ξ_1 with the edges E_{12} all carrying the fixed message distribution $\boldsymbol{\rho}^{2,1}(\boldsymbol{\mathsf{R}}, \boldsymbol{\delta}_\infty)$ in the constraint-to-variable direction. We refer to this as the *stability kernel* of the simple cascade. Without loss of generality, we can assume that each element of $\boldsymbol{\rho}^{2,1}(\mathbf{R}, \boldsymbol{\delta}_\infty)$ is equal to a received distribution.

Let Λ^s and $\mathcal{P}^s$ denote the stability matrix factors for stability kernel.

Theorem 8: If $\Lambda^s\big(\mathcal{B}(\mathbf{R})\big)\,\mathcal{P}^s$ is a stable matrix then there exists a constant $\gamma > 0$ such that

$$\nu(\mathbf{R}, \boldsymbol{\rho}(\mathbf{P}^\ell(\mathbf{Q}))) \xrightarrow{\ell\to\infty} \delta_\infty$$

for any $\mathbf{Q} \in \Psi(\mathbf{R})$ satisfying $\max_{i\in[1,j_2]}\{\mathrm{Pr}_{\mathrm{err}}(\mathsf{Q}_\mathsf{i})\} < \gamma$.

Proof: Assume $\Lambda^s\big(\mathcal{B}(\mathbf{R})\big)\mathcal{P}^s$ is stable. Consider now the slightly modified stability kernel with $n_\mathfrak{r}$ received types and j_1 edge types.

$$\nu^s(\mathbf{r}, \mathbf{r}', \boldsymbol{\delta}_\infty) := \frac{1}{\nu^1(\mathbf{1},\mathbf{1})}\nu^1(\mathbf{r}, \mathbf{x}', \mathbf{r}') \tag{18}$$

and μ^s defined as before. The original stability kernel sets each element of $\mathbf{r}'$ to a convex combination of elements of $\mathbf{r}$.

Let $\mathbf{H}_\eta$ denote the length $j_2 - j_1$ vector density where each element is the density associated to the BSC with cross-over probability η. Let $\mathbf{S}^\ell[\mathbf{H}_\eta]$ be the density evolution arising from the modified stability kernel with received distributions $\big(\mathbf{R}, \boldsymbol{\rho}^{2,1}(\mathbf{R}, \mathbf{H}_\eta)\big)$ and $\mathbf{S}^0 = \mathsf{Q}_{[1,j]}$. For all $\eta > 0$ sufficiently small the matrix $\Lambda^s\big(\mathcal{B}(\mathbf{R}, \boldsymbol{\rho}^{2,1}(\mathbf{R}, \mathbf{H}_\eta))\big)\mathcal{P}^s$ is stable. Hence, for all $\gamma > 0$ sufficiently small it follows that $\mathbf{S}^\ell[\mathbf{H}_\eta] \xrightarrow{\ell\to\infty} \boldsymbol{\delta}_\infty$. whenever $\min_i\{\mathsf{Q}_i\} \leq \gamma$.

Consider now the sequence $\mathbf{P}^\ell(\mathbf{Q})$. Assume that $\mathsf{P}^\ell_{[1,j_1]} \prec \mathbf{S}^\ell$ and $\mathsf{P}^\ell_{[j_1+1,j_2]} \prec \mathbf{H}_\eta$. It then follows that $\rho_{[j_1+1,\ldots,j_2]}(\mathbf{P}^\ell) \prec \boldsymbol{\rho}^{2,1}(\mathbf{R}, \mathbf{H}_\eta)$. From this we easily conclude that $\mathsf{P}^{\ell+1}_{[j_1+1,j_2]} = \lambda_{[j_1+1,j_2]}(\mathbf{R}, \boldsymbol{\rho}(\mathbf{P}^\ell)) \prec \mathbf{H}_\eta$, and that $\mathsf{P}^{\ell+1}_{[1,j_1]} = \lambda_{[1,j_1]}(\mathbf{R}, \boldsymbol{\rho}(\mathbf{P}^\ell)) \prec \mathbf{S}^{\ell+1}$. Since, for $\gamma \leq eta$ sufficiently small, we clearly have $\mathsf{P}^0_{[1,j_1]} \prec \mathbf{S}^0$ and $\mathsf{P}^0_{[j_1+1,j_2]} \prec \mathbf{H}_\eta$, we now conclude by induction that $\mathsf{P}^\ell_{[1,j_2]} \prec \mathbf{S}^\ell$, hence $\mathsf{P}^\ell_{[1,j_1]} \xrightarrow{\ell\to\infty} \boldsymbol{\delta}_\infty$. ■

X. Optimization

The main tool used to optimize structures is density evolution. Generally speaking, we use hill-climbing methods. Since the space of admissible parameters is a polytope, it is easy to generate extreme points using linear programming.

It is somewhat harder to explain how one arrives at good structures. This is something of an art. We have borrowed from the various examples in the literature and some ideas come out of interaction with the optimization. Many of the structural ideas presented here appear to be new, especially for low error floors.

There undoubtedly remain interesting structures to be uncovered.

XI. Multi-Edge type Examples

In this section we present some examples. The first few examples simply show how to express various well-known code structures in the multi-edge type framework. Later examples present new designs.

A. Standard Irregular LDPC

The standard irregular structure is represented as

$$\nu(r,x) = r_1 \sum_{j=2}^{J} \nu_j x_1^j \quad \text{and} \quad \mu(x) = \sum_{j=3}^{K} \mu_k x_1^k$$

where we abbreviate $\nu_{(0,1),j}$ as ν_j, with the received normalization $\sum_{j=2}^{J} \nu_j = 1$, the socket count constraint $\sum_{j=2}^{J} j\nu_j = \sum_{j=3}^{K} k\mu_k$, and the code rate $c = \sum_{j=2}^{J} \nu_j - \sum_{j=3}^{K} \mu_k$. From this we obtain

$$\lambda(r,x) = r_1 \sum_{j=2}^{J} \lambda_j x_1^{j-1} \quad \text{and} \quad \rho(x) = \sum_{j=3}^{K} \rho_k x_1^{k-1}$$

where we define $\lambda_k = \frac{j\nu_j}{\sum_{j=2}^{J} j\nu_j}$ and $\rho_k = \frac{k\mu_k}{\sum_{j=3}^{K} k\mu_k}$ The stability condition reduces to the familiar $\mathcal{B}(\mathsf{R}_1)\lambda_2\rho'(1) < 1$.

B. Repeat Accumulate (RA) Codes

Rate $1/q$ repeat accumulate codes are represented by a two edge type structure as follows.

$$\nu(r,x) = r_1 x_1^2 + \frac{1}{q} r_0 x_2^q \quad \text{and} \quad \mu(x) = x_1^2 x_2 \,.$$

Note that received constraint and the socket count constraints are satisfied. We obtain

$$\lambda(r,x) = \left(r_1 x_1, r_0 x_2^{q-1} \right) \quad \text{and} \quad \rho(x) = \left(x_1 x_2, x_1^2 \right)$$

and stability matrix is given by

$$\begin{bmatrix} \mathcal{B}(\mathsf{R}_1) & \mathcal{B}(\mathsf{R}_1) \\ 0 & 0 \end{bmatrix} .$$

Since $\mathcal{B}(\mathsf{R}_1) < 1$ (unless $\mathsf{R}_1 = \delta_0$ which we ruled out) we see that RA codes are guaranteed to be stable.

C. *Irregular Repeat Accumulate (IRA) Codes*

The IRA structure also requires two edge types as follows.

$$\nu(r,x) = \nu_{(2,0)} r_1 x_1^2 + \sum_{j=2}^{J} \nu_j r_1 x_2^j \quad \text{and} \quad \mu(x) = \sum_{j=3}^{K} \mu_k x_1^2 x_2^k$$

where we abbreviate $\nu_{(0,1),(2,0)}$ as $\nu_{(2,0)}$, $\nu_{(0,1),(0,j)}$ as ν_j and $\mu_{(2,j)}$ as μ_j. The received constraint is $1 = \nu_{(2,0)} + \sum_{j=2}^{J} \nu_j$, while the socket count constraints are $2\nu_{(2,0)} = 2\sum_{k=1}^{K} \mu_k$ and $\sum_{j=2}^{J} j\nu_j = \sum_{k=1}^{K} k\mu_k$. The code rate is given by $c = \nu_{(2,0)} + \sum_{j=2}^{J} \nu_j - \sum_{k=1}^{K} \mu_k = 1 - \sum_{k=1}^{K} \mu_k \; = 1 - \nu_{(2,0)}$. We obain

$$\lambda(r,x) = \left(r_1 x_1, r_1 \sum_{j=2}^{J} \lambda_j x_1^{j-1} \right) \quad \text{and} \quad \rho(x) = \left(x_1 \sum_{k=1}^{K} \frac{\mu_k}{1-c} x_2^k, x_1^2 \sum_{k=1}^{K} \rho_k x_2^{k-1} \right)$$

where we set $\lambda_j := \frac{j\nu_j}{\sum_{j=2}^{J} j\nu_j}$ and $\rho_k := \frac{k\mu_k}{\sum_{k=1}^{K} k\mu_k}$.

The stability matrix is computed as

$$\begin{bmatrix} \mathcal{B}(\mathsf{R}_1) & 0 \\ 0 & \mathcal{B}(\mathsf{R}_1)\lambda_2 \end{bmatrix} \begin{bmatrix} 1 & \sum_{k=1}^{K} \frac{k\mu_k}{1-c} \\ 2 & \sum_{k=1}^{K} (k-1)\rho_k \end{bmatrix} = \mathcal{B}(\mathsf{R}_1) \begin{bmatrix} 1 & \sum_{k=1}^{K} \frac{k\mu_k}{1-c} \\ 2\lambda_2 & \lambda_2 \sum_{k=1}^{K} (k-1)\rho_k \end{bmatrix}$$

Letting $\rho_2'(1)$ denote $\sum_{k=1}^{K} (k-1)\rho_k$ and $\rho_1'(1)$ denote $\sum_{k=1}^{K} \frac{k\mu_k}{1-c}$ we can write the eigenvalues of the stability matrix as

$$\mathcal{B}(\mathsf{R}_1) \frac{1 + \lambda_2 \rho_2'(1) \pm \sqrt{(1 - \lambda_2 \rho_2'(1))^2 + 8\lambda_2 \rho_1'(1)}}{2}$$

Note, in particular, that if $\lambda_2 = 0$ then stability is guaranteed.

The main difference between RA/IRA codes and standard irregular codes, from the structural perspective, is the control of degree two variable node connectivity. The structure allows many degree two nodes yet has guaranteed stability if there are no additional degree two nodes (with $\nu_2 = 0$ for IRA codes). It also simplifies encoding as the degree two nodes can be connected in a chain.

D. *CT codes*

The rate 1/2 (3,1) concatenated tree code structure [1] requires 5 edge types for its multi-edge type representation, see table I. Table I shows the basic structure for the $(3,1)$ CT-code [1]. The authors of [1] also propose certain encoding methods, graph constructions, and decoding schedules for these codes which are not considered here.

TABLE I

RATE 1/2 (3,1) CONCATENATED TREE CODE STRUCTURE

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$					$\mu_{\mathbf{d}}$	$\mathbf{d}$				
0.5	0	1	3	0	1	0	0	0.5	3	2	0	0	0
0.5	1	0	0	2	0	1	0	0.5	0	0	1	1	1
0.5	0	1	0	0	0	0	1						
AWGN: $\sigma^* = 0.925$													

The CT structure is a simple degree one cascade. The stability kernel, see table II, is interesting, because, in effect, one of the bits is repeated. Except for that feature, the stability kernel is an IRA structure. Thus, stability is guaranteed.

TABLE II

RATE 1/2 (3,1) STABILITY KERNEL FOR CONCATENATED TREE CODE STRUCTURE

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$		$\mu_{\mathbf{d}}$	$\mathbf{d}$	
0.5	0	2	3	0	0.5	3	2
0.5	0	1	0	2			

E. Constructions of Kantor and Saad

TABLE III

BASIC RATE 1/2 STRUCTURE OF KANTOR AND SAAD

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$			$\mu_{\mathbf{d}}$	$\mathbf{d}$		
0.500	1	0	5	0	0	0.050	1	2	0
0.500	0	1	0	2	0	0.450	2	2	0
0.500	0	1	0	0	1	0.075	6	0	1
						0.050	7	0	1
						0.375	2	0	1

The basic degree structure for the rate $1/2$ code over the AWGN proposed by Kantor and

TABLE IV

REFINED RATE 1/2 STRUCTURE OF KANTOR AND SAAD

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$				$\mu_{\mathbf{d}}$	$\mathbf{d}$			
0.450	1	0	2	3	0	0	0.050	1	0	2	0
0.050	1	0	1	4	0	0	0.450	2	0	2	0
0.500	0	1	0	0	2	0	0.075	0	6	0	1
0.500	0	1	0	0	0	1	0.050	0	7	0	1
							0.375	0	2	0	1
AWGN: $\sigma^* = 0$, but transition at $\sigma = 0.951$.											

Saad [5] is shown in table III. By construction, the variable nodes with degree $(5, 0, 0)$ are the information bits of the code. As given, the structure does not have a perfectly decodable fixed point, by Lemma 3. In their paper, Kantor and Saad say that edges should be connected uniformly. This suggests the modified structure indicated in table IV. Again, the structure does not have a perfectly decodable fixed point by Lemma 3. The problem is that this is essentially a simple cascade structure but it has degree 1 variable nodes in Ξ_1. No further control of connectivity can solve the problem. Nevertheless, density evolution shows that there is a fixed point with small error rates, near 5×10^{-4} averaged accross all variable nodes, and that a transition occurs to a fixed point with large error rate, near 10^{-1} for $\sigma \simeq 0.951$. If the isolated (degree 1 variable node) edges are spread out some improvement is possible, and the multi-edge type structure could model this with about 20 edge types, some improvement would be expected.

A small adjustment to the structure can turn it into a valid simple cascaded structure. Some doping of the systematic nodes is required. After making this modification and optimizing, we obtained the structure in table V which has a threshold of 0.952.

F. An optimized rate 1/2 example

Here we describe an example design of a degree structure for a rate 1/2 code for the AWGN. The structure is designed to have high threshold with low degrees, as a result, its performance for small block lengths is quite good. It is not designed to exhibit a deep error floor, and we will show how to modify the degree structure for deeper error floors later.

Table VI shows the basic degree structure. The AWGN threshold for this structure is about

TABLE V

Modified Rate 1/2 structure of Kantor and Saad

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$				$\mu_{\mathbf{d}}$	$\mathbf{d}$			
0.497124	1	0	2	3	0	0	0.499994	2	0	2	0
0.002865	0	1	2	3	0	0	0.060507	0	6	0	1
0.500000	0	1	0	0	2	0	0.052737	0	7	0	1
0.497135	0	1	0	0	0	1	0.383884	0	2	0	1
AWGN: $\sigma^* = 0.952$.											

TABLE VI

High threshold low degree rate $1/2$ multi-edge structure

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$				$\mu_{\mathbf{d}}$	$\mathbf{d}$			
0.5	0	1	2	0	0	0	0.4	4	1	0	0
0.3	0	1	3	0	0	0	0.1	3	2	0	0
0.2	1	0	0	3	3	0	0.2	0	0	3	1
0.2	0	1	0	0	0	1					
AWGN: $\sigma^* = 0.965$											

$\sigma^* = 0.965$. Further optimization of the parameters can increase that to about $\sigma^* = 0.968$. Table VII modifies the construction to put in degree two variable nodes in a chain, in the RA style. This imparts guaranteed stability to the structure. The threshold is reduced to $\sigma^* = 0.960$. Further optimization of the parameters can increase that to about $\sigma^* = 0.965$. Table VIII shows another structural modification to ensure that each puncture variable node receives some non-zero soft information in the first iteration. In this case, the structure uniquely determines the parameters. By introducing another variation on the edge split used to produce the structure and further optimizing, the threshold can be increased to $\sigma^* = 0.965$. Although each of these alterations slightly reduces the threshold, they impart other advantages.

Graphs generated from this structure outperform standard turbo codes down to block lengths of about 1000 and are significantly better for larger block lengths. This is a consequence of high thresholds (good asymptotic performance) with low degrees (fast convergence to the asymptote).

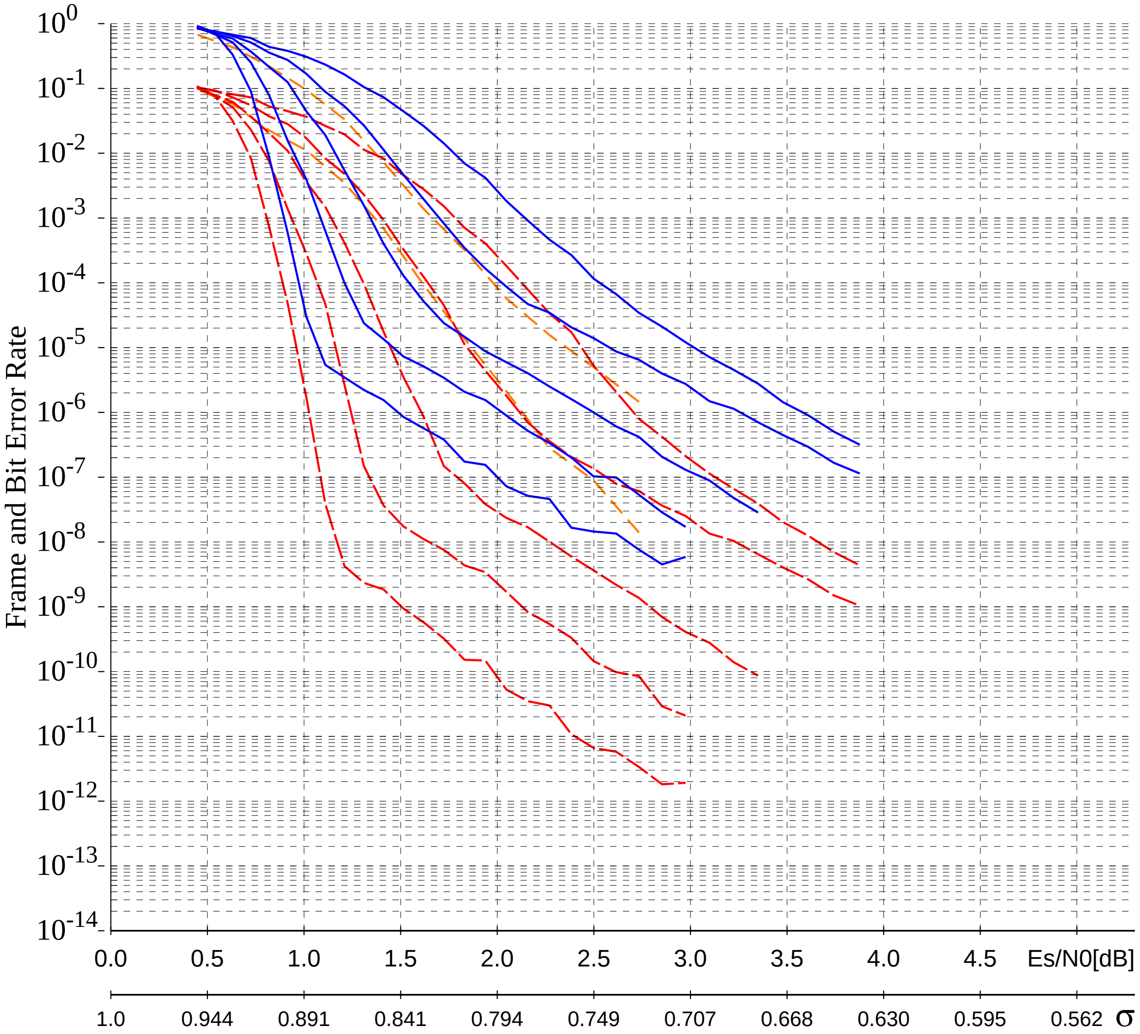

Fig. 1. Hardware based simulation for rate 1/2 codes with the degree structure given in Table VIII. Code parameters (n, k) are $(64x, 32x)$ with $x = 10, 20, 40, 80, 160$. Floating point simulations will yield improved performance with somewhat lower error floor. The dashed lines indicate floating point simulation results for the $x = 20$ code.

TABLE VII

MODIFICATION FOR GUARANTEED STABILITY OF $1/2$ MULTI-EDGE STRUCTURE

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$					$\mu_{\mathbf{d}}$	$\mathbf{d}$				
0.5	0	1	2	0	0	0	0	0.4	2	2	1	0	0
0.3	0	1	0	3	0	0	0	0.1	2	1	2	0	0
0.2	1	0	0	0	3	3	0	0.2	0	0	0	3	1
0.2	0	1	0	0	0	0	1						

AWGN: $\sigma^* = 0.960$

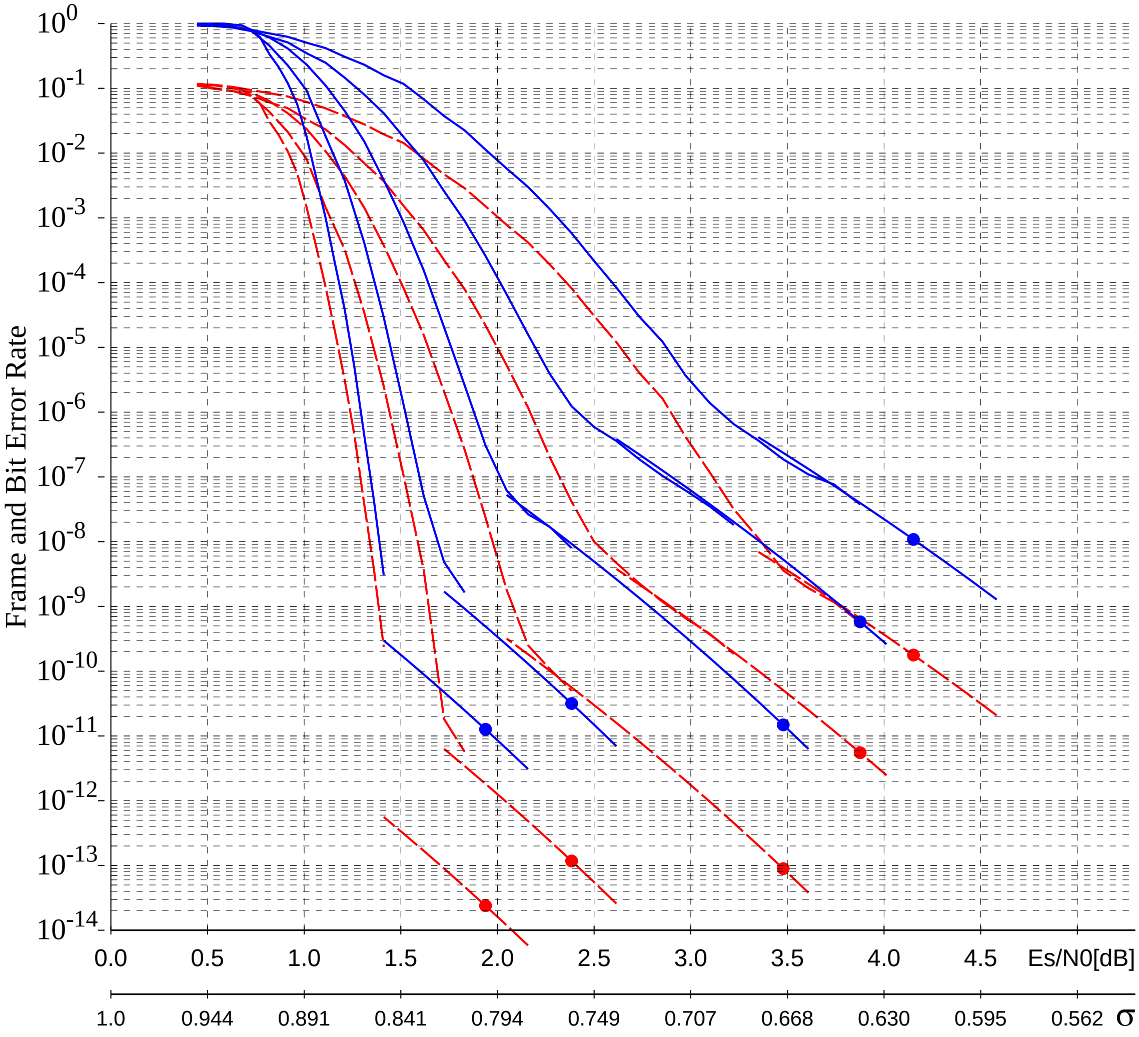

Fig. 2. Hardware based simulation for rate 1/2 codes with the degree structure given in Table IX. Code parameters (n, k) are $(64x, 32x)$ with $x = 10, 20, 40, 80, 160$. Compared to the performance in Fig. 1 we see about 0.2dB loss in the waterfall region, as predicted by the thresholds, and significantly better error floor performance. Since error floor simulation was not feasible in all cases, we also show error floor performance as predicted using the method outlined in [14].

TABLE VIII

Modification for rapid initialization of state variables of the rate$1/2$ multi-edge structure.

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$						$\mu_{\mathbf{d}}$	$\mathbf{d}$					
0.5	0	1	2	0	0	0	0	0	0.4	2	2	1	0	0	0
0.3	0	1	0	3	0	0	0	0	0.1	2	1	0	2	0	0
0.2	1	0	0	0	2	1	3	0	0.2	0	0	0	0	3	1
0.2	0	1	0	0	0	0	0	1							
AWGN: $\sigma^* = 0.960$															

TABLE IX

MODIFIED RATE 1/2 STRUCTURE FOR LOWER ERROR FLOOR.

Variable								Constraint					
$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$					$\mu_{\mathbf{d}}$	$\mathbf{d}$				
0.25	0	1	2	0	0	0	0	0.25	1	3	1	0	0
0.583333	0	1	0	3	0	0	0	0.25	1	4	1	0	0
0.166666	1	0	0	0	3	3	0	0.166666	0	0	0	3	1
0.166666	1	0	0	0	0	0	1						
AWGN: $\sigma^* = 0.936$													

G. LDPC for Low Error Floors

Recently, some advances have been made in our understanding of error floor behavior of LDPC codes [13], [14]. We will not reiterate that material here, but it is of use in developing the results as it provides insight into the structures that produce low error floors.

Let us first revisit the rate $\frac{1}{2}$ structure as presented in TableFloating point simulations will yield improved performance with somewhat lower error floor. VII. This structure produces graphs that have relatively high error floors. For the [10240,5120] code seen in Fig. 1 the error floor begins at about 4×10^{-9} BER and for the [5120,2560] code the error floor begins at about 5×10^{-8} BER (using a 5 bit hardware decoder). In this case, it is easy to understand that the relatively high error floor arises mostly from the large fraction of degree two variable nodes. By modifying the structure so as to reduce the number of degree two variable nodes, one can achieve much deeper error floors. Table IX shows one such modification. (This is not the higheset threshold possible, some modifications can yield a threshold of $\sigma^* = 0.947$). Constraint nodes have at most one degree two neighbor. For the [5120,2560] code the error floor begins at about 2×10^{-11} BER and for the [10240,5120] code the error floor is predicted to begin at about 4×10^{-13} BER, see Fig. 2.

More interesting effects can be found by looking at high rate codes. This has practical importance since many applications requiring high speeds and low error floors use high rate codes. We will present here only an indication of what can be done. Consider a regular (3,9) structure.

$$\nu(r,x) = r_1 x_1^3 \qquad \text{and} \qquad \mu(x) = \frac{1}{3} x_1^9 \,.$$

This is a rate 2/3 code with AWGN threshold $\sigma^* \simeq 0.708$. Error floors arise because of small unavoidable structures in the graph. If the right degree could be lowered then the error floor could be lowered. By introducing high degree punctured nodes this can be accomplished. In this case, and in many cases, it can also result in an *improvement* in the threshold. Consider a regular (3,8) structure but to every check node attach one more edge going to a degree 9 puntured variable node. The resulting degree structure is the following.

$$\nu(r,x) = r_1 x_1^3 + \frac{1}{24} r_0 x_2^9 \qquad \text{and} \qquad \mu(x) = \frac{3}{8} x_1^8 x_2 \,. \tag{19}$$

The resulting code is still rate 2/3 but the error floor is that that arises from the (3,8) structure. The degee 9 variable nodes essentially do not participate in error floor failure events. Perhaps surprisingly, the threshold for the AWGN is $\sigma^* \simeq 0.721$, better than for the regular case. Some simulation results are presented in Fig. 3.

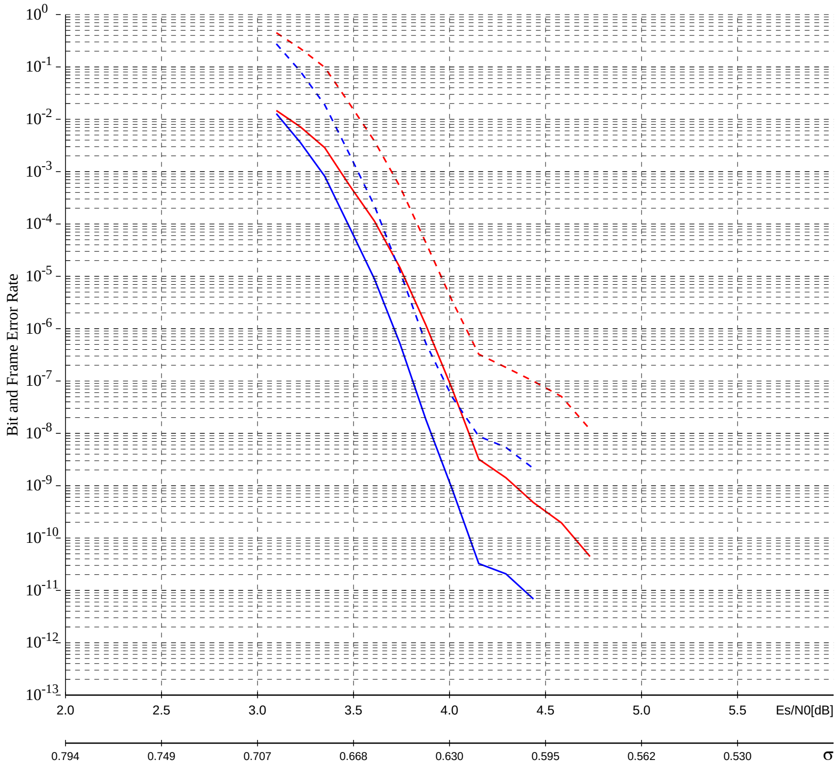

Fig. 3. Rate 2/3 examples. BER and FER (dashed) curves for regular (3,9) code and MET code (19). Note the MET code outperforms the regular one both in the waterfall and error floor region. Simulation performed on hardware platform using a 5 bit decoder.

The idea of puncturing nodes to maintain girth was suggested by MacKay [15]. Note that the idea here is quite different, the girth is not improved. On the (3,8) portion of the graph the girth distribution is improved but on the overall graph it is not. The fact that the punctured nodes are high degree and that their connectivity is controlled is critical to achieving the results presented here.

H. Low Rate LDPC

One encounters a difficulty when designing very low rate LDPC codes in the standard irregular framework, it is relatively difficult to achieve high thresholds. In contrast, for high rate codes, with rates approaching 1, even regular codes have quite good thresholds. It turns out that simple cascades provide an ideal framework for low rate LDPC codes. One begins with a high rate code, and then appends to it a large number of parity check bits. An example for a simple rate $1/10$ degree structure is given in table X. The AWGN threshold for this structure is $\sigma^* \simeq 2.535$, the Shannon limit is 2.6, leaving a gap of about 0.2 dB. This structure is somewhat similar to the recently introduced Raptor codes [16].

TABLE X

RATE 1/10 DEGREE STRUCTURE

$\nu_{\mathbf{b},\mathbf{d}}$	$\mathbf{b}$		$\mathbf{d}$			$\mu_{\mathbf{d}}$	$\mathbf{d}$		
0.1	0	1	3	20	0	0.025	15	0	0
0.025	0	1	3	25	0	0.875	0	3	1
0.875	0	1	0	0	1				
AWGN: $\sigma^* = 2.535$									

XII. CONCLUSIONS

The multi-edge type framework is a natural generalization of the notion of irregularity of LDPC codes. This framework captures most degree structures of interest in an efficient and compact language. Through some examples we have illustrated several techniques that be used to improve various aspects of LDPC performance.

ACKNOWLEDGEMENTS

The first author would like to thank Hui Jin for useful discussions and Vladimir Novichkov for implementing the hardware used to perform the simulations.

REFERENCES

[1] L.Ping and K. Y. Wu, “Concatenated tree codes: A low complexity high performance approach,” *IEEE Trans. on Information Theory*, vol. 47, pp. 791–800, Feb. 2001.

[2] D. Divsalar, H. Jin, and R. McEliece, “Coding theorems for ”turbo-like” codes,” in *Proceedings of the 1998 Allerton Conference*, p. 210, 1998.

[3] H. Jin, A. Khandekar, , and R. J. McEliece, “Irregular repeat accumulate codes,” in *2nd International Symposium on Turbo Codes and Related Topics*, (Brest, France), pp. 1–8, Sept. 2000.

[4] D. J. C. MacKay and R. M. Neal, “Good codes based on very sparse matrices,” in *Cryptography and Coding. 5th IMA Conference* (C. Boyd, ed.), no. 1025 in Lecture Notes in Computer Science, pp. 100–111, Berlin: Springer, 1995.

[5] I. Kantor and D. Saad, “Error-correcting codes that nearly saturate Shannon’s bound,” *Phys.Rev.Lett.*, vol. 83, pp. 2660–2663, 1999.

[6] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” *IEEE Trans. on Information Theory*, vol. 47, pp. 599–619, Feb. 2001.

[7] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity approaching low-density parity-check codes,” *IEEE Trans. on Information Theory*, vol. 47, pp. 599–619, Feb. 2001.

[8] H. Jin and T. Richardson, “On iterative joint decoding and demodulation,” in *Allerton Conf. on Communication, Control and Computing*, (Monticello, Illinois), Oct. 2003.

[9] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Improved low-density parity-check codes using irregular graphs,” *IEEE Trans. on Information Theory*, vol. 47, pp. 585–598, Feb. 2001.

[10] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept. 1981.

[11] T. Richardson and R. Urbanke, “Fixed points and stabilty of density evolution.” in preparation.

[12] H. Jin and T. Richardson, “Fast density evolution,” in *Proceedings CISS*, (Princeton, NJ), Mar. 2004.

[13] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, “Finite-length scaling for iteratively decoded ldpc ensembles,” in *Allerton Conf. on Communication, Control and Computing*, (Monticello, Illinois), Oct. 2003.

[14] T. Richardson, “Error floors of ldpc codes,” in *Allerton Conf. on Communication, Control and Computing*, (Monticello, Illinois), Oct. 2003.

[15] D. J. C. MacKay, “Punctured and irregular high rate Gallager codes.” preprint available at http://www.inference.phy.cam.ac.uk/mackay/puncture.pdf.

[16] M. Shokrollahi, “Raptor codes.” preprint, 2002.

APPENDIX

Asymptotics of Distributions Let R by a symmetric density. Define

$$\hat{\mathsf{R}}(\omega) := \int_{-\infty}^{\infty} e^{-i\omega x} e^{-\frac{1}{2}x} \mathsf{R}(x)\, dx\,.$$

Note that this is the Fourier transform evaluated at $\frac{1}{2}+i\omega$ and that $\hat{\mathsf{R}}$ is real and even $\hat{\mathsf{R}}(-\omega) = \hat{\mathsf{R}}(\omega)$. By symmetry we have $\frac{d}{d\omega}\hat{\mathsf{R}}(\omega) \mid_{\omega=0}= 0$. Now

$$\begin{aligned}\frac{d^2}{d\omega^2}\hat{\mathsf{R}}(\omega) &= -\int_{-\infty}^{\infty} x^2 e^{-i\omega x} e^{-\frac{1}{2}x}\mathsf{R}(x)\,dx \\ &= -2\int_0^{\infty} x^2\cos(\omega x)e^{-\frac{1}{2}x}\mathsf{R}(x)\,dx\end{aligned}$$

Since $\int_0^\infty \mathsf{R}(x)\,dx \le 1$ it now follows that

$$\begin{aligned}\frac{d^2}{d\omega^2}\hat{\mathsf{R}}(\omega) &\geq -2\max_{x\in[0,\infty)} x^2\,|\cos(\omega x)|\,e^{-\frac{1}{2}x} \\ &\geq -2\max_{x\in[0,\infty)} x^2 e^{-\frac{1}{2}x} \\ &= -32e^{-2}\end{aligned}$$

Noting that $\hat{\mathsf{R}}(0) = \mathcal{B}(\mathsf{R})$, it now follows that

$$\hat{\mathsf{R}}(\omega) \geq \mathcal{B}(\mathsf{R})\big(1-\frac{1}{\mathcal{B}(\mathsf{R})}16e^{-2}\omega^2\big)$$

Assume ℓ is even. Then $\hat{\mathsf{R}}^\ell(\omega) \geq 0$. We can now write

$$\hat{\mathsf{R}}^\ell(\omega) \geq \left(\mathcal{B}(\mathsf{R})\big(1-\frac{1}{\mathcal{B}(\mathsf{R})}16e^{-2}\omega^2\big)^+\right)^\ell \geq \mathcal{B}(\mathsf{R})^\ell(1-\frac{\ell}{\mathcal{B}(\mathsf{R})}16e^{-2}\omega^2)^+$$

Applying Plancherel's theorem, we have for any distribution R,

$$\begin{aligned}\Pr_{\text{err}}(\mathsf{R}) &= \frac{1}{2}\int_{-\infty}^{\infty} e^{-|\frac{1}{2}x|}e^{-\frac{1}{2}x}\mathsf{R}(x)\,dx \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{1}{1+4\omega^2}\hat{\mathsf{R}}(\omega)\,dw \\ &= \frac{2}{\pi}\int_0^{\infty}\frac{1}{1+4\omega^2}\hat{\mathsf{R}}(\omega)\,dw\end{aligned}$$

Assume again that ℓ is even. We now have

$$\begin{aligned}
\mathrm{Pr}_{\mathsf{err}}(\mathsf{R}^{\otimes\ell}) &= \int_0^\infty \frac{2/\pi}{1+4\omega^2}\hat{\mathsf{R}}^\ell(\omega)\,dw \\
&\geq \mathcal{B}(\mathsf{R})^\ell \int_0^{\frac{e}{4}\sqrt{\frac{\mathcal{B}(\mathsf{R})}{\ell}}} \frac{2/\pi}{1+4\omega^2}(1-\frac{\ell}{\mathcal{B}(\mathsf{R})}16e^{-2}\omega^2)dw \\
&\geq \mathcal{B}(\mathsf{R})^\ell \frac{2/\pi}{1+\frac{e^2\mathcal{B}(\mathsf{R})}{4\ell}} \int_0^{\frac{e}{4}\sqrt{\frac{\mathcal{B}(\mathsf{R})}{\ell}}} (1-\frac{\ell}{\mathcal{B}(\mathsf{R})}16e^{-2}\omega^2)dw \\
&= \mathcal{B}(\mathsf{R})^\ell \frac{2/\pi}{1+\frac{e^2\mathcal{B}(\mathsf{R})}{4\ell}} \frac{2}{3}\frac{e}{4}\sqrt{\frac{\mathcal{B}(\mathsf{R})}{\ell}} \\
&= \mathcal{B}(\mathsf{R})^\ell \frac{1}{1+\frac{e^2\mathcal{B}(\mathsf{R})}{4\ell}} \frac{e}{3\pi}\sqrt{\frac{\mathcal{B}(\mathsf{R})}{\ell}}
\end{aligned}$$

For the case ℓ is odd, we note that $\frac{1}{\mathcal{B}(\mathsf{R})}\hat{\mathsf{R}}(\omega) \leq 1$. We can then write

$$\begin{aligned}
\mathrm{Pr}_{\mathsf{err}}(\mathsf{R}^{\otimes\ell}) &= \int_0^\infty \frac{2/\pi}{1+4\omega^2}\hat{\mathsf{R}}^\ell(\omega)\,dw \\
&\geq \frac{1}{\mathcal{B}(()\mathsf{R})}\int_0^\infty \frac{2/\pi}{1+4\omega^2}\hat{\mathsf{R}}^{\ell+1}(\omega)\,dw
\end{aligned}$$

and proceed as before.

So, in general, we have

$$\mathrm{Pr}_{\mathsf{err}}(\mathsf{R}^{\otimes\ell}) \geq \mathcal{B}(^)\ell(\mathsf{R})\frac{1}{1+\frac{e^2\mathcal{B}(\mathsf{R})}{4(\ell+1)}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}(\mathsf{R})}{\ell+1}}$$

We want to generalize this to the matrix case. Consider the matrix $M(\mathbf{R}) := \Lambda(\mathbf{R})\mathcal{P}$. Each entry of $M(\mathbf{R})$ is of the form $\sum_{i=1}^{n_\mathfrak{r}} a_i\mathsf{R}_i$ where the a_i are positive coefficients. Thus, the entries of $M^\ell(\mathbf{R})$ are of the form $\sum a_i\mathbf{R}^{\mathbf{d}^i}$ where the a_i are positive coefficients and $\sum_{j=1}^{n_\mathfrak{r}} \mathbf{d}_j^i = \ell$. Let $\mathcal{B}_{\min} := \min\mathcal{B}(()\mathsf{R}_i)$. Applying essentially the same argument as above we obtain

$$\mathrm{Pr}_{\mathsf{err}}(\mathbf{R}^{\mathbf{d}^i}) \geq \mathcal{B}(\mathbf{R}^{\mathbf{d}^i})\frac{1}{1+\frac{e^2\mathcal{B}_{\min}}{4(\ell+n_\mathfrak{r})}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}_{\min}}{\ell+n_\mathfrak{r}}}$$

hence

$$\begin{aligned}
\mathrm{Pr}_{\mathsf{err}}(M(\mathbf{R})^\ell) &\geq \frac{1}{1+\frac{e^2\mathcal{B}_{\min}}{4(\ell+n_\mathfrak{r})}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}_{\min}}{\ell+n_\mathfrak{r}}}\mathcal{B}(M(\mathbf{R}))^\ell \\
&= \frac{1}{1+\frac{e^2\mathcal{B}_{\min}}{4(\ell+n_\mathfrak{r})}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}_{\min}}{\ell+n_\mathfrak{r}}}\,(\,M(\mathcal{B}(\mathbf{R}))\,)^\ell
\end{aligned}$$

where the inequality is component-wise. If v is a positive eigenvector of $M(\mathcal{B}(\mathbf{R}))$ with positive eigenvalue ξ, then

$$\mathrm{Pr}_{\mathrm{err}}(M(\mathbf{R})^{\ell}v) \geq \frac{1}{1+\frac{e^2\mathcal{B}_{\min}}{4(\ell+n_{\mathfrak{r}})}}\frac{e}{3\pi}\sqrt{\frac{\mathcal{B}_{\min}}{\ell+n_{\mathfrak{r}}}}\,\xi^{\ell}v$$

component-wise. ■

List of Figures

List of Tables