Design of Provably Good Low-Density Parity Check Codes

Tom Richardson, Amin Shokrollahi and Rüdiger Urbanke
Bell Labs, Lucent Technologies
Murray Hill, NJ 07974

April 5, 1999

Abstract

We design sequences of low-density parity check codes that provably perform at rates extremely close to the Shannon capacity. The codes are built from highly irregular bipartite graphs with carefully chosen degree patterns on both sides. Our theoretical analysis of the codes is based on [1]. Additionally, based on the assumption that the underlying communication channel is symmetric, we prove that the probability densities at the message nodes of the graph satisfy a certain symmetry. This enables us to derive a succinct description of the density evolution for the case of a belief propagation decoder. Furthermore, we prove a stability condition which implies an upper bound on the fraction of errors that a belief propagation decoder can correct when applied to a code induced from a bipartite graph with a given degree distribution.

Our codes are found by optimizing the degree structure of the underlying graphs. We develop several strategies to perform this optimization. We also present some simulation results for the codes found which show that the performance of the codes is very close to the asymptotic theoretical bounds.

Index Terms – Low density parity check codes, belief propagation, turbo codes, irregular codes

1 Introduction

In this paper we present *irregular* low-density parity check codes (LDPCCs) which exhibit performance extremely close to the best possible as determined by the Shannon capacity formula. For the additive white Gaussian noise channel (AWGNC) the best code of rate one-half presented in this paper has a threshold within 0.06dB from capacity, and simulation results show that our best LDPCC of length 10^6 achieves a bit error probability of 10^{-6} less than 0.13dB away from capacity, handily beating the best (turbo) codes known so far.

LDPCCs possess several other distinct advantages over turbo-codes. First, the complexity of (belief-propagation) decoding is somewhat less than that of turbo-codes and, being fully parallelizable, can potentially be performed at significantly greater speeds. Second, as indicated in a previous paper [1], very low complexity decoders that closely approximate belief-propagation in performance may be (and have been) designed for these codes. Third, low-density parity check decoding is verifiable in the sense that decoding to a correct codeword is a detectable event. One practical objection to low-density parity-check codes has been that their encoding complexity is high. One way to get around this problem is by slightly modifying the construction of codes from bipartite graphs to a cascade of such graphs, see [2, 3]. A different solution for practical purposes, which does not require cascades, will be presented elsewhere [4].

Let us recall some basic notation. As is well known, low-density parity-check codes are well represented by bipartite graphs in which one set of nodes, the *variable nodes*, corresponds to elements of the codeword and the other set of nodes, the *check nodes*, corresponds to the set of parity-check constraints satisfied by codewords of the code. *Regular* low-density parity-check codes are those for which all nodes of the same type have the same degree. Thus, a (3,6)-regular low-density parity-check code has a graphical representation in which all variable nodes have degree 3 and all check nodes have degree 6. The bipartite graph determining such a code is shown in Fig. 1.

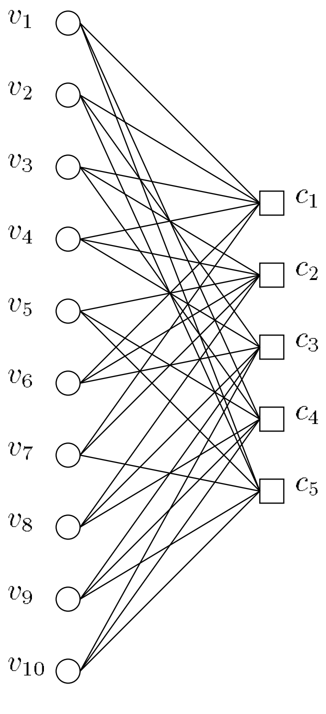

Figure 1: A $(3,6)$-regular code of length 10. There are 10 variable nodes and 5 check nodes. For each check node c_i the sum (over GF(2)) of all adjacent variable nodes is equal to zero.

For an *irregular* low-density parity-check code the degrees of each set of nodes are chosen according to some distribution. Thus, an irregular low-density parity-check code might have a graphical representation in which half the variable nodes have degree 5 and half have degree 3, while half the constraint nodes have degree 6 and half have degree 8. For a given length and a given degree sequence (finite distribution) we define an *ensemble* of codes by choosing the edges, i.e., the connections between variable and check nodes, randomly. More precisely, we enumerate the edges emanating from the variable nodes in some arbitrary order and proceed in the same way with the edges emanating from the check nodes. Assume that the number of edges is E. Then a code (a particular instance of this ensemble) can be identified with a permutation on E letters. Note that all elements in this ensemble are equiprobable. In practice the edges are not chosen entirely randomly since certain potentially unfortunate events in the graph construction can be easily avoided.

In a recent paper [1] we presented an asymptotic analysis of LDPCCs under message passing decoding. Assume we have the following setup:

1. We are given an ordered family of binary-input discrete memoryless channels parameterized by a real parameter δ such that if $\delta_1 < \delta_2$ then the channel with parameter δ_2 is a physically degraded version of the channel with parameter δ_1. Further, each channel in this family fulfills the channel symmetry condition

$$p(y|x=1) = p(-y|x=-1)\ . \tag{1}$$

2. A sequence of code ensembles $\mathcal{C}_n(\lambda, \rho)$ is specified, where n is the length of the code and $\lambda(x) := \sum_{i=1}^{d_l} \lambda_i x^{i-1}$ ($\rho(x) := \sum_{i=1}^{d_r} \rho_i x^{i-1}$) is the variable (check) node degree sequence. More precisely, λ_i (ρ_i) is the fraction of edges emanating from variable (check) nodes of degree i. Note that the rate of the code is given in terms of $\lambda(x)$ and $\rho(x)$ as $1 - \frac{\int_0^1 dx\, \rho(x)}{\int_0^1 dx\, \lambda(x)}$.

3. A message passing decoder is selected. By definition, messages only contain *extrinsic* information, i.e., the message emitted along an edge e does not depend on the incoming message along the same edge. Further, the decoder fulfills the following symmetry conditions. Flipping the sign of all incoming messages at a variable node results in a flip of the sign of all outgoing messages. The symmetry condition at a check node is slightly more involved. Let e be an edge emanating from a check node c. Then flipping the sign of i incoming messages arriving at node c, excluding the message along edge e, results in a change of the sign of the outgoing message by $(-1)^i$. In all these cases, only the sign is changed, but the reliability remains unchanged.

Under the above assumptions there exists a threshold δ^* with the following properties: For any $\epsilon > 0$ and $\delta < \delta^*$ there exists an $n(\epsilon, \delta)$ such that almost every code[1] in $\mathcal{C}_n(\lambda, \rho)$, $n > n(\epsilon, \delta)$, has probability of error smaller than ϵ assuming that transmission takes place over a channel with parameter δ. Conversely, if the transmission takes place over a channel with parameter $\delta > \delta^*$, then almost any code in $\mathcal{C}_n(\lambda, \rho)$ has probability of error uniformly bounded away from zero.[2]

Further, for the important case of belief propagation decoders a procedure was introduced that enables the efficient computation of δ^* to any desired degree of accuracy. In [1] threshold values and simulation results were given for a variety of noise models, but the class of low-density parity-check codes considered was largely restricted to *regular* codes. In the present paper we present results indicating the remarkable performance that can be achieved by properly chosen *irregular* codes.

The idea underlying this paper is quite straightforward. Assume we are interested in transmission over a particular memoryless channel using a particular message passing decoder. Since for every given pair of degree sequences (λ, ρ) we can determine the resulting threshold value δ^*, it is natural to search for those pairs which maximize this threshold.[3] This was done, with great success, in the case of erasure channels [5, 6, 7]. In the case of most other channels of interest the situation is much more complicated and new methods must be brought to bear on the optimization problem. Fig. 2 compares the performance of the $(3, 6)$-regular LDPCC (which is the best regular such code) with the performance of the best irregular LDPCC we found in our search and the performance of the standard parallel turbo code introduced by Berrou, Galvieux, and Thitimajshima [8]. All three codes have rate one-half and their performance under belief propagation decoding over the AWGNC is shown for a code word length 10^6. Also shown is the Shannon limit and the threshold value of our best LDPCC ($\sigma^* = 0.9718$). From this figure it is clear how much benefit can be derived from optimizing the degree sequences. For $n = 10^6$ and a bit error probability of 10^{-6}, our best LDPCC is only 0.13dB away from capacity. This handily beats the performance of turbo-codes. Even more impressive, the threshold, which indicates the performance for infinite lengths, is a mere 0.06dB away from capacity.

[1] More precisely, the fraction of codes for which the above statement is true converges exponentially (in n) fast to 1.

[2] We conjecture that this is actually true for *every* code in $\mathcal{C}_n(\lambda, \rho)$.

[3] We may also optimize degree sequences under various constraints. For example, the larger the degrees used the larger the code size needs to be in order to approach the predicted asymptote. Therefore, it is highly desirable to look for the best degree sequences with some a priori bound on the size of the degrees.

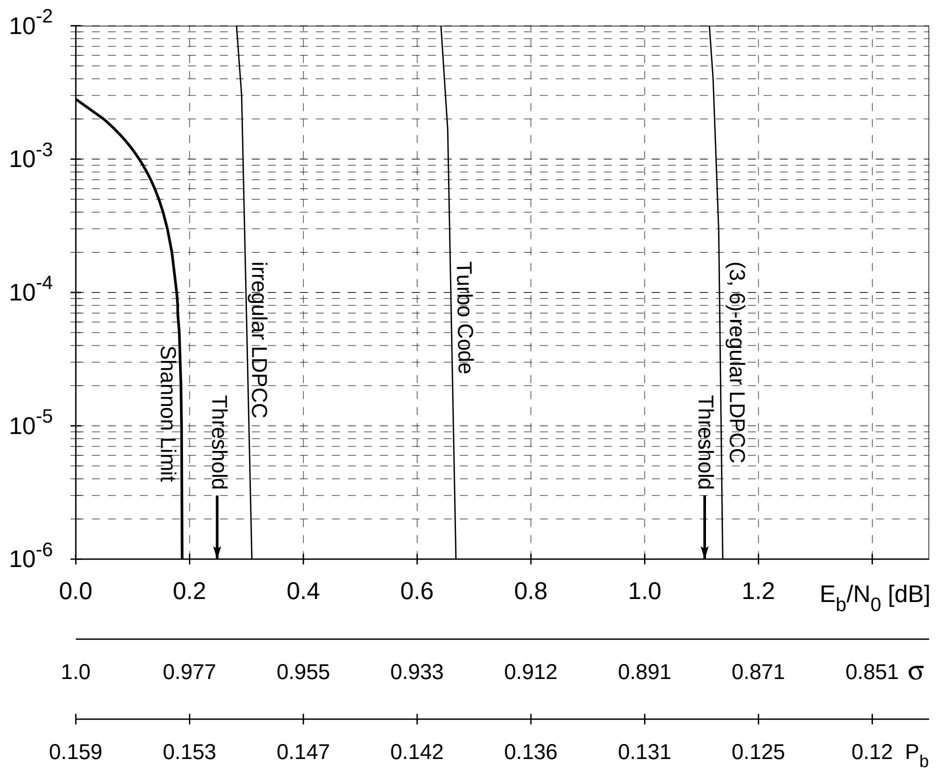

Figure 2: Comparison of $(3,6)$-regular LDPCC, turbo code, and optimized irregular LDPCC. All codes are of length 10^6 and of rate one-half. The bit error rate for the AWGNC is shown as a function of E_b/N_0 (in dB), the standard deviation σ, as well as the raw input bit error probability P_b.

The empirical evidence presented in Fig. 2 together with the results presented in Section 3 beg the question of whether LDPCCs can achieve capacity on a given binary-input memoryless channel. The only result in this direction is that of [5] which gives an explicit sequence of degree distributions such that, in the limit, the codes induced by these degree distributions achieve capacity on an erasure channel. The following Theorem, due to Gallager, imposes, at least for the BSC, a necessary condition in order for LDPCCs to achieve capacity: their maximum right degree d_r must tend to infinity.[4] Although this result bounds the performance of LDPCCs away from capacity, the gap is extremely small and the gap converges to zero exponentially fast in d_r. Hence, although of great theoretical interest, the following theorem does not impose a significant practical limitation.

Theorem 1. [Gallager 61] Let $\mathcal{C} \in \mathcal{C}(\lambda, \rho)$ be a LDPCC of rate r. Let $\mathcal{C}$ be used over a BSC with crossover probability δ and assume that each codeword is used with equal probability. If

[4] We conjecture that a similar statement (and proof) could be given for continuous channels.

$r > 1 - h(\delta)/h(p^*)$, where $h(\cdot)$ is the binary entropy function and $p^* = \frac{1+(1-2\delta)^{d_r}}{2}$, then the block error probability is bounded away from 0.

Proof. Note that the capacity of the BSC is equal to $1 - h(\delta)$. Since $p^* > 1/2$ for any finite d_r, we have $h(p^*) < 1$ and, hence, $1 - h(\delta)/h(p^*) < 1 - h(\delta)$. Gallager stated the above Theorem originally for (j,k)-regular codes. A closer look, however, shows that the proof remains valid for the case of irregular codes if the original expression $p^* = \frac{1+(1-2\delta)^k}{2}$ is replaced by the expression $p^* = \frac{1+(1-2\delta)^{d_r}}{2}$. Details of the proof can be found in [9, p. 39]. We note that a slightly sharper bound can be given if we replace d_r with the average degree of the $(1-r)$-fraction of highest degree nodes. However, since the improvement is usually only slight and since the exact size of the gap is not significant in practice, we leave the details to the reader. $\square$

The outline of this paper is as follows. In Section 2 we describe and study density evolution. Under the assumption that the input distribution arises from a symmetric channel, we show that the message distributions satisfy a certain *consistency* condition which is invariant under density evolution. Many simplifications arise in this case which afford us considerable information on the nature of density evolution. In particular, we show that density evolution always converges to a fixed point. We also address the stability of the fixed point that corresponds to the correct solution. In Section 3 we give tables of some very good degree sequences. Although we focus mostly on the AWGNC and rates one-half, we also give a few examples for different channels and rates. In Section 3 we also present some simulation results that show that the promised performance can be closely achieved for reasonably long codes. Finally, in Section 4 we describe our numerical optimization techniques which were used to generate the tables in Section 3.

2 Density Evolution: Consistency, Stability, and Fixed Points

Density evolution, as described in [1], is general enough to enable the computation of message densities regardless of the initial density. In the case of most interest, the message distribution arises from the observation of the input bit after passing it through a known symmetric channel. Since the channel is known, the log-likelihood for the input bit given the observation is determined and this log-likelihood constitutes the received message. Message distributions (under the ubiquitous all-one word assumption) which arise this way satisfy certain consistency conditions that are preserved under density evolution. In the following, we will examine these conditions. They will lead to simplifications of density evolution. Furthermore, in this case, one can prove a monotonicity property of density evolution which implies that density evolution converges to a fixed point. Finally, in this section, we study the asymptotics of correct decoding to arrive at a stability condition for the fixed point corresponding to zero probability of error.

Let us first briefly recall the form and derivation of density evolution for belief propagation. In general, each message represents an estimate of a particular transmitted bit. Let x denote the value of this bit and recall that x is an element of $\{\pm 1\}$. At a variable node, we will represent messages as log-likelihood ratios,

$$\log \frac{p(y|x=1)}{p(y|x=-1)} ,$$

where y represents all the observations conveyed to the variable node at that time. In the sequel we will often refer to message distributions without further qualifications. Unless stated otherwise, this will always refer to message distributions at variable nodes under the all-one word assumption where the messages are represented as log-likelihood ratios. At a check node it is more convenient to represent the message as a tuple (s, r) where $s \in W_2 := \{\pm 1\}$ indicates the hard decision and where $r \in \mathbb{R}^+$ is equal to

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

Here W_2 denotes the two-element group with elements $\{\pm 1\}$ and group operation equal to real multiplication. Since we wish to admit the possibility of perfect knowledge of a bit (erasure channel), we must admit the possibility that message distributions may have point masses at the endpoints. E.g., in the log-likelihood representation this means that we may have 'point masses at infinity' in our message distributions. Thus, in general, message distributions are probability distributions on the two-point compactification of $\mathbb{R}$, which we denote by $\bar{\mathbb{R}}$. To simplify the presentation we shall generally assume that distributions are absolutely continuous with respect to Lebesgue measure and can therefore be represented by their densities.[5]

Assume now that the received message and the incoming messages at a variable node are given in log-likelihood form. The outgoing message along an edge e is then simply the sum of the received message plus all incoming messages, excluding the message incoming along edge e. Therefore,

[5]A couple of exceptions will be made for certain discrete measures. In particular, we shall require the distribution Δ_∞, the 'delta function at infinity.' We shall also require the distribution Δ_0 the 'delta function at zero,' corresponding to an erasure.

assuming independence (tree assumption), the *density* of the outgoing message is the convolution of the densities of the messages participating in this sum.

An equivalent statement is true at the check nodes. Assume that the incoming messages to a check node are all in the form (s, r). The outgoing message (again in (s, r) representation) along an edge e is then simply the sum of all incoming messages, excluding the message incoming along edge e (here "addition" is performed componentwise, i.e., $(s_1, r_1) + (s_2, r_2) = (s_1 * s_2, r_1 + r_2)$, where the first component is in $W_2 = \{\pm 1\}$ and "addition" corresponds to real multiplication, and the second component is an element of $\mathbb{R}$ with regular addition.) Thus, as in the previous case, the density of the outgoing message, over $W_2 \times \bar{\mathbb{R}}^+$, can be written as the convolution of the corresponding densities of the incoming messages. Here, $\bar{\mathbb{R}}^+$ refers to $[0, \infty]$ the compactification of $\mathbb{R}^+$. To complete the description we need only specify the change of variables necessary to convert a density from one representation to the other and to incorporate the degree sequences.

Suppose that we have a graph with left and right edge degree distributions given by $\lambda(x) = \sum_{i=1}^{d_\ell} \lambda_i x^{i-1}$, and $\rho(x) = \sum_{i=1}^{d_r} \rho_i x^{i-1}$, respectively. We follow [3] and use the following convention: for a polynomial $q(x) = \sum_i q_i x^{i-1}$ with real coefficients q_i we denote by $q(f)$ the distribution $\sum_i q_i f^{\otimes(i-1)}$, where here and in the sequel $\otimes$ denotes convolution.

In the following, we assume that the channel satisfies the symmetry condition (1). Further, we assume that the all-one codeword was transmitted. Let P_0 be the distribution of the received log-likelihoods and P_ℓ denote the distribution of log-likelihoods sent from the variable nodes to the check nodes in the ℓth iteration. Let R_ℓ denote the distribution of log-likelihoods sent from the check nodes to the variable nodes in the ℓth iteration. We may initialize with $R_0 = \Delta_0$.

The distribution of messages passed from a message node to a check node in the ℓth iteration is given by the convolution $P_\ell := P_0 \otimes \lambda(R_{\ell-1})$.

To obtain the distribution of messages from check nodes back to message nodes, we must perform a change of measure. To this end we define the operator γ which takes the space of distributions on $\bar{\mathbb{R}}^+$ into itself. For densities on $\mathbb{R}^+$ we define γ as

$$\forall\, y \geq 0: \qquad \gamma(f)(y) := f(\ln \coth y/2)\,\mathrm{csch}(y). \tag{2}$$

The operator γ represents a change of variables $y \to \ln \coth y/2$, i.e., $f(y)\,\mathrm{d}y = \gamma(f)(y)\,\mathrm{d}y$, since $\frac{\mathrm{d}}{\mathrm{d}y} \ln \coth y/2 = -\,\mathrm{csch}(y)$. (We can extend the definition of γ to distributions on $\bar{\mathbb{R}}^+$ by taking limits.) This shows part (a) of the following lemma, while part (b) follows easily from the fact that

$$\ln\coth\left(\frac{\ln\coth y/2}{2}\right) = y\,.$$

Lemma 1. *The following facts hold:*

(a) $\int_0^\infty \gamma(f)(y)\,\mathrm{d}y = \int_0^\infty f(y)\,\mathrm{d}y$

(b) $\gamma(\gamma(f))(y) = f(y)$.

For any distribution f on $\bar{\mathbb{R}}$ let f^- denote $1_{[-\infty,0]}\,f$ and let f^+ denote $1_{[0,\infty]}\,f$, where 1_A denotes the characteristic function of A so that $f = f^+ + f^-$.[6]

Given a distribution f over $\bar{\mathbb{R}}$ of log-likelihoods l, we can represent it as a distribution over $W_2 \times \bar{\mathbb{R}}^+$ by representing l as the pair (s, r). We therefore define Γ, the change of variable operator, by

$$\Gamma(f)(s,r) = 1_{\{1\}}(s)\gamma(f^+)(r) + 1_{\{-1\}}(s)\gamma(\mathcal{R}f^-)(r)\,.$$

where $\mathcal{R}f^-$ denotes the reflection of f about 0, i.e., $\mathcal{R}f^-(x) = f^-(-x)$.

Now, let g be a distribution over $W_2 \times \bar{\mathbb{R}}^+$. Let g^+ and g^- be defined by $g^+(r) = 1_{\{1\}}(s)g(s,r)$ and $g^-(r) = 1_{\{-1\}}(s)g(s,r)$. Then the inverse of Γ is given by

$$\Gamma^{-1}(g)(l) = \gamma(g^+)(l) + \mathcal{R}\gamma(g^-)(l)\,.$$

Using this notation, we obtain the following.

Theorem 2 (Density Evolution). *Let P_0 denote the initial message distribution, under the assumption that the all-one codeword was transmitted, of a low-density parity check code specified by edge degree distributions $\sum_i \lambda_i x^{i-1}$ and $\sum_i \rho_i x^{i-1}$. If R_ℓ denotes the density of the messages passed from the check nodes to the message nodes at round ℓ of the belief-propagation, with $R_0 := \Delta_0$, then we have*

$$R_\ell = \Gamma^{-1}\rho(\Gamma(P_0 \otimes \lambda(R_{\ell-1})))\,.$$

2.1 Consistency

In the following, we call a distribution f on $\bar{\mathbb{R}}$ *consistent* if it satisfies $f(x) = f(-x)\mathrm{e}^x$ for all $x \in \bar{\mathbb{R}}^+$. More formally, a probability measure ν is consistent if the following hold. First, ν can

[6] In the general case, where a point mass at 0 is allowed, the mass at 0 should be split equally between f^- and f^+.

be written as $\nu^- + \nu^+$ where ν^- is supported on $[-\infty, 0]$ and ν^+ is supported on $[0, +\infty]$. Letting $\mathcal{R}\nu^-$ denote as before the reflection of ν^- about 0, we require that $\mathcal{R}\nu^-$ be absolutely continuous with respect to ν^+ and that the Radon-Nikodyn derivative [10, p. 180] of $\mathcal{R}\nu^-$ with respect ν^+ be given by e^{-x}. If ν has a density f then consistency of ν is equivalent to consistency of f as defined above. Note that Δ_∞ and Δ_0 are consistent measures.

We will first establish consistency of the initial message distribution of density evolution under general conditions.

Proposition 1. *Suppose that a channel has symmetry property $p(y \mid x = 1) = p(-y \mid x = -1)$, and let P_0 be the initial message distribution in log-likelihood ratio form under the all-one word assumption. Then P_0 is consistent.*

Proof. Let ν denote the measure corresponding to P_0, i.e., for any measurable subset A, $\nu(A) := \int_A P_0(x)\ \mathrm{d}x$. Note that P_0 is equal to the distribution of the random variable

$$z(y) := \log \frac{p(y|x = 1)}{p(y|x = -1)} \tag{3}$$

under the assumption that the all-one codeword was transmitted. For any measurable set A we have $\nu(A) = \int_{z^{-1}(A)} p(y|x = 1)\ \mathrm{d}y$. Invoking channel symmetry and equation (3) we have

$$\begin{aligned}
\nu(-A) &= \int_{z^{-1}(-A)} p(y|x = 1)\ \mathrm{d}y \\
&= \int_{z^{-1}(A)} p(-y|x = 1)\ \mathrm{d}y \\
&= \int_{z^{-1}(A)} p(y|x = -1)\ \mathrm{d}y\,. \\
&= \int_{z^{-1}(A)} \mathrm{e}^{-z(y)} p(y|x = 1)\ \mathrm{d}y\,.
\end{aligned}$$

Thus, we obtain that ν is a consistent measure, i.e., setting $A = (z, z + \mathrm{d}z)$ we obtain

$$\frac{\nu(-z - \mathrm{d}z, -z)}{\nu(z, z + \mathrm{d}z)} = \mathrm{e}^{-z}\,.$$

□

Example 1 (Erasure Channel). For the erasure channel the initial message distribution is $\epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$. Then $P_0^+ = \frac{\epsilon}{2}\Delta_0 + (1-\epsilon)\Delta_\infty$ and $P_0^- = \frac{\epsilon}{2}\Delta_0$, which shows that the initial distribution is consistent.

□

Example 2 (BSC). For the BSC the initial message distribution is $P_0(x) := \delta\Delta_{-\log\frac{1-\delta}{\delta}} + (1-\delta)\Delta_{\log\frac{1-\delta}{\delta}}$. We have $P_0^+ = (1-\delta)\Delta_{\log\frac{1-\delta}{\delta}}$ and $P_0^- = \delta\Delta_{\log\frac{1-\delta}{\delta}}$, so consistency is apparent.

$\square$

Example 3 (AWGNC). Here the initial message distribution is $P_0(x) := \sqrt{\frac{\sigma^2}{8\pi}}\mathrm{e}^{-\frac{(x-\frac{2}{\sigma^2})^2\sigma^2}{8}}$. The consistency condition is then

$$P_0(x) = \sqrt{\frac{\sigma^2}{8\pi}}\mathrm{e}^{-\frac{(x-\frac{2}{\sigma^2})^2\sigma^2}{8}} = \sqrt{\frac{\sigma^2}{8\pi}}\mathrm{e}^{-\frac{(-x-\frac{2}{\sigma^2})^2\sigma^2}{8}}\mathrm{e}^x = P_0(-x)\mathrm{e}^x.$$

$\square$

Example 4 (Laplace Channel). As a final example, the initial message distribution for the Laplace channel is given by

$$P_0(x) = \frac{1}{2}\Delta_{\frac{2}{\lambda}} + \frac{1}{2}e^{-\frac{2}{\lambda}}\Delta_{-\frac{2}{\lambda}} + \frac{1}{4}e^{\frac{x-\frac{2}{\lambda}}{2}}\chi_{|x|\le\frac{2}{\lambda}},$$

where

$$\chi_{|x|\le\frac{2}{\lambda}} := \begin{cases} 1, & |x| \le \frac{2}{\lambda}, \\ 0, & |x| > \frac{2}{\lambda}. \end{cases}$$

Again, it is easy to check that the consistency condition is fulfilled.

We say that two channels are *equivalent* if they have the same initial message distribution. It is often convenient to pick one representative for each equivalence class. How this can be done is shown in

Lemma 2. Let $P_0(y)$ be a consistent distribution. Then the symmetric channel $p(\cdot|\cdot)$ with $p(y|x = 1) = P_0(y)$ (and hence by symmetry $p(y|x=-1) = P_0(-y)$) has initial message distribution $P_0(y)$.

Proof.

$$\log\frac{p(y|x=1)}{p(y|x=-1)} = \log\frac{p(y|x=1)}{p(-y|x=1)} = \log\frac{P_0(y)}{P_0(-y)} = \log\frac{P_0(y)}{P_0(y)\mathrm{e}^{-y}} = y.$$

$\square$

Our next goal is to show that consistency is preserved under density evolution. This leads to certain simplifications of its representation.

Lemma 3. *A distribution f over $\bar{\mathbb{R}}$ is consistent if and only if $\gamma(\mathcal{R}f^-)(y) = \tanh(y/2)\gamma(f^+)(y)$.*

Proof. We have

$$\gamma(\mathcal{R}f^-)(y) = f^-(-\ln\coth(y/2))\,\mathrm{csch}(y)$$

and

$$\gamma(f^+)(y) = f^+(\ln\coth(y/2))\,\mathrm{csch}(y)\,.$$

Note that from $f(-x) = f(x)\mathrm{e}^{-x}$ it follows that

$$\gamma(\mathcal{R}f^-)(y) = \exp(-\ln\coth(y/2))f^+(\ln\coth(y/2))\,\mathrm{csch}(y) = \tanh(y/2)\gamma(f^+)(y)\,,$$

and the lemma follows directly. □

Thus, we may extend the definition of consistency to densities over $W_2 \times \bar{\mathbb{R}}^+$.

Lemma 4. *Consistency is invariant under convolution in* $\bar{\mathbb{R}}$ *and convolution in* $W_2 \times \bar{\mathbb{R}}^+$.

Proof. Let f and g be consistent densities over $\bar{\mathbb{R}}$. Then

$$\begin{aligned}\int_{-\infty}^{\infty} f(x-y)g(y)\,\mathrm{d}y &= \int_{-\infty}^{\infty} \mathrm{e}^{x-y}f(y-x)\mathrm{e}^{y}g(-y)\,\mathrm{d}y \\ &= \int_{-\infty}^{\infty} \mathrm{e}^{x}f(y-x)g(-y)\,\mathrm{d}y \\ &= \mathrm{e}^{x}\int_{-\infty}^{\infty} f(-x-y)g(y)\,\mathrm{d}y\,.\end{aligned}$$

To show that consistency is invariant under convolution over $W_2 \times \bar{\mathbb{R}}^+$ is somewhat more elaborate. Let f and g be consistent densities over $W_2 \times \bar{\mathbb{R}}^+$ and let us express them as

$$\begin{aligned} f(s,r) &= 1_{\{1\}}(s)f^+(r) + 1_{\{-1\}}(s)f^-(r) \\ g(s,r) &= 1_{\{1\}}(s)g^+(r) + 1_{\{-1\}}(s)g^-(r)\,,\end{aligned}$$

(where $f^+(r) = 1_{\{1\}}(s)f(s,r)$, etc.) and let us similarly express $f \otimes g$. Then the convolution of f and g is specified by

$$\begin{aligned} (f\otimes g)^+ + (f\otimes g)^- &= (f^+ + f^-)\otimes(g^+ + g^-) \\ (f\otimes g)^+ - (f\otimes g)^- &= (f^+ - f^-)\otimes(g^+ - g^-)\,.\end{aligned}$$

Now, the consistency condition for a distribution f over $W_2 \times \bar{\mathbb{R}}^+$ can be expressed as

$$(f^+ - f^-)(y) = \mathrm{e}^{-y}(f^+ + f^-)(y)\,.$$

Thus, it is easily verified that $f \otimes g$ is consistent. □

In view of the above it is convenient to introduce the "fold" operator $\mathcal{F}$ which takes distributions on $\bar{\mathbb{R}}$ to distributions on $\bar{\mathbb{R}}^+$. We define it by

$$\mathcal{F}f(x) = f^+(x) + \mathcal{R}f^-(x) = f^+(x) + f^-(-x)\,.$$

Altogether, we obtain the following.

Theorem 3 (Density Evolution and Consistency). *Let P_0 denote the initial message distribution of a low-density parity check code specified by edge degree distributions $\sum_i \lambda_i x^{i-1}$ and $\sum_i \rho_i x^{i-1}$ and assume that P_0 is consistent. If R_ℓ denotes the density of the messages passed from the check nodes to the message nodes at round ℓ of the belief-propagation then we have*

$$R_\ell = (\mathcal{F}^{-1} \circ \gamma)\rho((\gamma \circ \mathcal{F})(P_0 \otimes \lambda(R_{\ell-1}))),$$

and P_ℓ is consistent for all $\ell \geq 0$.

Example 5 (Erasure Channel). Let f be a density distribution of the form $\epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$. Then $\mathcal{F}(f) = f$, and $\gamma(f) = (1-\epsilon)\Delta_0 + \epsilon\Delta_\infty$. Further, $\lambda(f) = \lambda(\epsilon)\Delta_0 + (1-\lambda(\epsilon))\Delta_\infty$. Hence, starting with the initial message distribution $\epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$ corresponding to an erasure channel with erasure probability ϵ, the iteration is given by $P_\ell = p_\ell\Delta_0 + (1-p_\ell)\Delta_\infty$, where

$$p_\ell = \rho(1 - \epsilon\lambda(1-p_{\ell-1})).$$

This is the same formula as derived in [5, 6].

$\square$

Let $f(x)$ be a consistent distribution. We define the error probability operator as[7]

$$\mathrm{Pr}_{\mathrm{err}}(f) := \int_{-\infty}^{0} f(x)\ \mathrm{d}x\ .$$

An easy but helpful consequence of the consistency condition is noted in the following

Corollary 1. Let $f(x)$ be a consistent distribution. If $\mathrm{Pr}_{\mathrm{err}}(f) = 0$ then $f = \Delta_\infty$.

Hence, requiring that the probability of error converges to zero is equivalent to requiring that the message density converges to a delta function at infinity.

[7]If f has a point mass at 0 then exactly half of that mass is included in $\mathrm{Pr}_{\mathrm{err}}(f)$.

2.2 Fixed Points of Density Evolution

In the case of the erasure channel and a belief propagation decoder the state of the system at any iteration is described by the remaining number of erasures. Denote the remaining number of erasures by x and let $h(x)$ be the remaining number of erasures after a further iteration. Then the threshold is given by the maximum number x_0 such that

$$\forall 0 < x < x_0 : \quad h(x) < x. \tag{4}$$

A similar statement is true for the BSC together with Gallager's decoding algorithm A. In this case x signifies the remaining number of errors.

An alternative description of the threshold can be given in terms of fixed points. The threshold is given by the maximum number x_0 such that for no $x \in (0, x_0)$

$$h(x) = x \ . \tag{5}$$

Assume now we are given a general discrete memoryless channel which fulfills the channel symmetry conditions and we employ a belief propagation decoder. The state of the system is now described by the message density. Let f denote such a density and let $h(f)$ denote the corresponding density after a further iteration. Clearly, there is no characterization of the threshold which corresponds to (4) since there is, a priori, no given linear ordering of densities. The alternative formulation in (5) involving fixed points looks more promising. Unfortunately, the non-existence of fixed-points for iterated function systems in more than one dimension is in general not enough to guarantee convergence. So it is quite surprising that for message distributions of belief propagation decoders fixed points sufficiently characterize the convergence of the sequences, as we will show in this section.

Let $P_\ell(x)$ denote the distribution of the messages at the ℓ-th iteration assuming that the all-one word was transmitted. Assume that we were to decide on the transmitted bit according to the sign of the message. In this case the conditional error probability, conditioned on the transmission of the all-one codeword, is equal to $\Pr_{\rm err}(P_\ell)$. But, because of the symmetry conditions, this is equal to the unconditional error probability. Since the sign of the message is equal to the MAP estimate, this error probability is clearly a non-increasing function of ℓ.

This monotonicity property will play a key role in the sequel. It is not hard to see that there is actually a whole family of such monotonicity conditions.

Theorem 4. *Let P_ℓ be the message distribution at the ℓ-th decoding step and let g be a consistent distribution on $\bar{\mathbb{R}}$. Then*

$$\mathrm{Pr}_{err}(P_\ell \otimes g)$$

is a non-increasing function of ℓ.

Proof. The message of which P_ℓ is the distribution represents a conditional probability of a particular bit value. Assume that an independent observation of the same bit value is available to the decoder and assume that this independent observation is obtained by passing the bit through a channel $p(\cdot|\cdot)$ which fulfills the symmetry condition and has $p(y|x=1) = g(y)$. By Lemma 2 and under the assumption that the all-one codeword was transmitted, the conditional distribution of the bit value log-likelihood, conditioned on all information incorporated in P_ℓ and the independent observation, has distribution $P_\ell \otimes g$. Since the new distribution corresponds again to a MAP estimate conditioned on information which is non-decreasing in ℓ, the stated monotonicity condition follows. $\square$

It is convenient to use the following "basis" for all these monotonicity conditions. Let

$$g_z(x) := \frac{1}{1+\mathrm{e}^z}\Delta_{-z} + \frac{\mathrm{e}^z}{1+\mathrm{e}^z}\Delta_z \,. \tag{6}$$

Clearly, $g_z(x)$ is a consistent distribution. For any consistent distribution f let $\mathcal{P}_z f$ be defined as

$$\mathcal{P}_z f := \mathrm{Pr}_{\mathrm{err}}(f \otimes g_z).$$

Written in an alternative way we have

$$\mathcal{P}_z f := \frac{1}{1+\mathrm{e}^z}\int_0^z \mathrm{d}x\; f(x)(1+\mathrm{e}^{-x}) + \int_z^\infty \mathrm{d}x\; f(x)\mathrm{e}^{-x} \;=\frac{1}{1+\mathrm{e}^z}\left(1-\int_z^\infty (1-\mathrm{e}^{z-x})f(x)\,\mathrm{d}x\right). \tag{7}$$

Corollary 2. *Let P_ℓ be the message distribution at the ℓ-th decoding step. Then $\mathcal{P}_z P_\ell$ is a non-increasing function of ℓ for every $0 \le z \le \infty$.*

Lemma 5 (Basis Lemma). *Let f and g be consistent distributions such that $\mathcal{P}_z g = \mathcal{P}_z f$, $\forall z \ge 0$. Then $f = g$.*

Proof. Write the conditions $\mathcal{P}_z(f(x) - g(x)) = 0$, $\forall z \ge 0$, in the equivalent form

$$\forall z \ge 0: \quad \int_z^\infty (f(x) - g(x))(1-\mathrm{e}^{z-x})\,\mathrm{d}x = 0\,.$$

Differentiating both sides with respect to z we get

$$\forall z \geq 0 : \int_z^\infty (f(x) - g(x))\mathrm{e}^{-x} \, \mathrm{d}x = 0$$

or, equivalently,

$$\forall z \geq 0 : \int_{-\infty}^{-z} (f(x) - g(x)) \, \mathrm{d}x = 0 \, .$$

Clearly, the last condition implies $f(x) = g(x)$, for all $x \leq 0$. Since f and g are both distributions, so that $\int_{-\infty}^{\infty}(f(x) - g(x))\,\mathrm{d}x = 0$, and since f and g are both consistent it follows that $f = g$. $\square$

Theorem 5. Let f be a consistent distribution and assume that f has support over the whole real axis. Let $P_{\ell_1}(x)$ and $P_{\ell_2}(x)$ denote the message distributions at the ℓ_1-th and ℓ_2-th iteration respectively, with $\ell_1 < \ell_2$. If

$$\mathrm{Pr}_{\mathrm{err}}(P_{\ell_1} \otimes f) = \mathrm{Pr}_{\mathrm{err}}(P_{\ell_2} \otimes f) \tag{8}$$

then $P_{\ell_1}(x) = P_\ell(x)$ for $\ell \geq \ell_1$, i.e., $P_{\ell_1}(x)$ is a fixed point of density evolution.

Proof. We first establish the following identity for any consistent distribution f.

$$f(z) = \int_0^{+\infty} \mathcal{F}f(x)\, g_x(z) \, \mathrm{d}x \tag{9}$$

The identity is proved as follows.

$$\begin{aligned} f(z) &= \int_{-\infty}^{+\infty} f(x)\, \Delta_0(z - x) \, \mathrm{d}x \\ &= \int_0^{+\infty} \mathcal{F}f(x) \left(\frac{1}{1 + e^{-x}} \Delta_0(z - x) + \frac{e^{-x}}{1 + e^{-x}} \Delta_0(-z - x) \right) \mathrm{d}x \\ &= \int_0^{+\infty} \mathcal{F}f(x)\, g_x(z) \, \mathrm{d}x \, . \end{aligned}$$

Now, let ℓ satisfy $\ell_1 \leq \ell \leq \ell_2$. Applying the identity above we have

$$P_\ell \otimes f = \int_0^{+\infty} \mathcal{F}f(x)\, (g_x \otimes P_\ell)(z) \, \mathrm{d}x \, ,$$

from which it follows that

$$\mathrm{Pr}_{\mathrm{err}}(P_\ell \otimes f) = \int_0^{+\infty} \mathcal{F}f(x)\, \mathcal{P}_x P_\ell \, \mathrm{d}x \, .$$

Note that (8) together with Theorem 4 implies that

$$\mathrm{Pr}_{\mathrm{err}}(P_\ell \otimes f) = \mathrm{Pr}_{\mathrm{err}}(P_{\ell_1} \otimes f) \, .$$

Thus, we have

$$0 = \int_0^{+\infty} \mathcal{F}f(x)\,(\mathcal{P}_x P_{\ell_1} - \mathcal{P}_x P_\ell)\ \mathrm{d}x\,.$$

It is clear from (7) that $\mathcal{P}_x P_\ell$ is a continuous function of x. Further, f has support on $\mathbb{R}$ by assumption and since, by Corollary 2, $\mathcal{P}_x P_\ell$ is non-increasing in ℓ and, hence, the whole integrand is a non-negative function, it follows that $\mathcal{P}_x P_{\ell_1} = \mathcal{P}_x P_\ell$ for all $0 \le x < \infty$. By the Basis Lemma this proves that $P_{\ell_1} = P_\ell$. In particular we have $P_{\ell_1} = P_{\ell_1+1}$ so P_{ℓ_1} is a fixed point of density evolution. □

We note that the above theorem has some very strong implications. Although the message distributions for a belief propagation decoder live nominally on $\mathbb{R}^{\mathbb{R}}$, the above theorem implies that the possible message distributions can be ordered by one real parameter (the projection). Hence we are dealing with a one-dimensional manifold embedded in an infinite dimensional space. This is very much equivalent to the simple one-dimensional case we encounter in the case of an erasure channel, or Gallager's decoding algorithm A.

2.3 Stability

In [11] it was observed that a necessary condition for successful termination of the decoding process in case of an erasure channel with erasure probability ϵ is $\lambda'(0)\rho'(1) < 1/\epsilon$. This condition can be interpreted as a stability condition for the fixed point 0 of the iteration procedure given in Example 5. In this section we will derive similar conditions for other types of channels.

Observe that if $P_\ell = \Delta_\infty$ for some $\ell \ge 0$ then $P_{\ell+i} = \Delta_\infty$ for any $i \ge 0$, i.e., Δ_∞ is a fixed point of density evolution. Since we desire that the error probability associated with the density evolution converges to zero, it is clear that the fixed point described above should, in some sense, be an attractor (recall Corollary 1). To analyze local convergence to this fixed point we shall consider a linearization of density evolution about the fixed point.

Consider a density $\epsilon P + (1-\epsilon)\Delta_\infty$ where P is any (consistent) density, $P \neq \Delta_\infty$. After a complete iteration of density evolution this density will evolve to

$$\lambda'(0)\rho'(1)\epsilon P \otimes P_0 + (1 - \lambda'(0)\rho'(1)\epsilon)\Delta_\infty + O(\epsilon^2)\,,$$

where $\lambda'(x)$ and $\rho'(x)$ denote the derivatives of $\lambda(x)$ and $\rho(x)$, respectively. In other words, the linearization of density evolution at the fixed point Δ_∞ is given by $P \to \lambda'(0)\rho'(1)P \otimes P_0$.

One would expect that a necessary condition for convergence to zero of the probability of error be that the probability of error associated with

$$(\lambda'(0)\rho'(1))^n P \otimes P_0^{\otimes n}$$

be decaying to zero as n grows. Under very general conditions (see Lemma 7) the limit

$$r := -\lim_{n\to\infty} \frac{1}{n} \log \mathrm{Pr}_{\mathrm{err}}(P_0^{\otimes n}) \tag{10}$$

is well defined. In this case, since $P \neq \Delta_\infty$, the quantity $(\lambda'(0)\rho'(1))^n \mathrm{Pr}_{\mathrm{err}}(P \otimes P_0^{\otimes n})$ converges to zero if and only if $(\lambda'(0)\rho'(1))^n \mathrm{Pr}_{\mathrm{err}}(P_0^{\otimes n})$ converges to zero. Before proving this fact we will prove the following useful result.

Lemma 6. *Let p be a symmetric channel and let P be its associated message distribution. Then the channel p can be represented as the concatenation of an erasure channel with erasure probability $2\mathrm{Pr}_{err}(P)$ and a symmetric channel q, i.e., p is a physical degraded version of an erasure channel.*

Proof. Recall from Lemma 2 that without loss of generality we may assume that $p(y|x=1) = P(y)$ and hence, by symmetry, $p(y|x=-1) = P(-y)$. Let ε denote an erasure and let $1, -1$ denote bit values. Let q denote a channel whose input alphabet is $\{-1, \varepsilon, 1\}$ with probabilities, $(1-2\epsilon)/2, 2\epsilon, (1-2\epsilon)/2$ respectively, i.e., the input to q is the output of the binary erasure channel with erasure probability 2ϵ. Let the channel q have real output y and set $q(y|x=\varepsilon) = \frac{1}{2\epsilon} e^{-|y|} P(|y|)$, $q(y|x=1) = \frac{1}{1-2\epsilon}(1-e^{-y})P^{+}(y)$, and $q(y|x=-1) = \frac{1}{1-2\epsilon}(1-e^{y})P^{+}(-y)$. It is easy to check that these quantities are well defined distributions, e.g., integrate to one. Let $\tilde{q}$ denote the concatenation of the erasure channel with the channel q. Then

$$\tilde{q}(y|x=1) = 2\epsilon \frac{1}{2\epsilon} e^{-|y|} P(|y|) + (1-2\epsilon)\frac{1}{1-2\epsilon}(1-e^{-y})P^{+}(y) = P(y) = p(y|x=1)\,,$$

and

$$\tilde{q}(-y|x=1) = 2\epsilon \frac{1}{2\epsilon} e^{-|y|} P(|y|) + (1-2\epsilon)\frac{1}{1-2\epsilon}(1-e^{y})P^{+}(-y) = P(-y) = p(y|x=-1)\,.$$

□

Since P is consistent we have

$$\mathrm{Pr}_{\mathrm{err}}(P \otimes P_0^{\otimes n}) \leq \mathrm{Pr}_{\mathrm{err}}(P_0^{\otimes n})\,.$$

Let $\epsilon := \Pr_{\text{err}}(P)$. It follows from Lemma 6 that

$$\Pr_{\text{err}}(P \otimes P_0^{\otimes n}) \geq \Pr_{\text{err}}((2\epsilon\Delta_0 + (1-2\epsilon)\Delta_\infty) \otimes P_0^{\otimes n}) = 2\epsilon \Pr_{\text{err}}(P_0^{\otimes n})\,.$$

This shows that P effects the expression $(\lambda'(0)\rho'(1))^n \Pr_{\text{err}}(P \otimes P_0^{\otimes n})$ at most by a constant factor 2ϵ and, hence, to decide on the limiting behavior we can look at $(\lambda'(0)\rho'(1))^n \Pr_{\text{err}}(P_0^{\otimes n})$ instead.

Theorem 6. *If $\lambda'(0)\rho'(1) > \mathrm{e}^r$, then the probability of error of density evolution is strictly bounded away from* 0.

Proof. Let γ denote $\lambda'(0)\rho'(1)$ and assume that $\gamma > e^r$. Then, there exists an integer n such that

$$\gamma^n \int_{-\infty}^{0} P_0^{\otimes n}(x)\, \mathrm{d}x > 1\,.$$

Let density evolution be initialized with a consistent message density P satisfying $\Pr_{\text{err}}(P) = \epsilon$ and let P_n denote the message distribution obtained after n iterations of density evolution. (Here P_0 still denotes the received density while the message density is initialized to P.) Let $\tilde{P}_n$ denote the message distribution obtained after n iterations with the initial message distribution set instead to to $2\epsilon\Delta_0 + (1-2\epsilon)\Delta_\infty$.

Consider a random tree of depth $2n+1$ with variable nodes at the leaves and a variable node at the root. Let the leaf variable nodes have observations from a channel corresponding to the initial message distribution Q and let the other variable nodes have observations from a channel corresponding to P_0. Then, P_n and $\tilde{P}_n$ are the message distributions out of the root node corresponding to $Q = P$ and $Q = 2\epsilon\Delta_0 + (1-2\epsilon)\Delta_\infty$ respectively. Since the messages are conditional likelihoods it now follows from Lemma 6 that

$$\Pr_{\text{err}}(P_n) \geq \Pr_{\text{err}}(\tilde{P}_n)\,.$$

It is easy to see that

$$\Pr_{\text{err}}(\tilde{P}_n) = 2\epsilon\gamma^n \int_{-\infty}^{0} P_0^{\otimes n}(x)\, \mathrm{d}x + O(\epsilon^2)\,.$$

For ϵ sufficiently small it follows that $\Pr_{\text{err}}(\tilde{P}_n)$, and hence $\Pr_{\text{err}}(P_n)$, exceeds 2ϵ.

Since the probability of error associated to density evolution (when the message distribution is initialized to P_0) is non-increasing it follows that the probability of error can never reach ϵ. □

Note that the proof shows that the fixed point Δ_∞ is unstable. For any consistent initial message distribution the probability of error diverges away from 0 if the conditions of the theorem are met.

We note that in all cases of interest the exponent r can be computed using moment generating functions as stated in the following

Lemma 7. *Let $g(s)$ be the moment generating function corresponding to the distribution $P_0(x)$, i.e., $g(s) = E_{P_0}[\mathrm{e}^{sX}]$, and assume that $g(s) < \infty$ for all s in some neighborhood of zero. Then $r = -\log\left(\inf_{s<0} g(s)\right)$. Further, if P_0 is consistent then $r = -\log\left(2\int_0^\infty P_0^+(x)\,\mathrm{e}^{-x/2}\,\mathrm{d}x\right)$.*

Proof. A general large deviation principle (see, e.g., [12]) implies that $r = \inf_{s<0} g(s)$. In case that P_0 is consistent we have

$$\begin{aligned}
\mathrm{e}^{-r} &= \inf_{s<0} g(s) \\
&= \inf_{s<0} \int_{-\infty}^{\infty} P_0(x)\,\mathrm{e}^{sx}\,\mathrm{d}x \\
&= \inf_{s<0} \int_{0}^{\infty} P_0^+(x)\,\mathrm{e}^{-x/2}[\mathrm{e}^{-x(s+\frac{1}{2})} + \mathrm{e}^{x(s+\frac{1}{2})}]\,\mathrm{d}x \\
&= 2\int_0^\infty P_0^+(x)\mathrm{e}^{-x/2}\,\mathrm{d}x.
\end{aligned}$$

□

Example 6 (Erasure Channel). *For the erasure channel (see Example 1) we have*

$$\mathrm{e}^{-r} = 2\int_0^\infty [\frac{\epsilon}{2}\Delta_0 + (1-\epsilon)\Delta_\infty]\,\mathrm{e}^{-x/2}\,\mathrm{d}x = \epsilon\,.$$

Therefore, the stability condition reads

$$\lambda'(0)\rho'(1) < \frac{1}{\epsilon},$$

as previously obtained in [11].

Example 7. [BSC] For the BSC (see Example 2) we have

$$\mathrm{e}^{-r} = 2\int_0^\infty (1-\delta)\Delta_{\log\frac{1-\delta}{\delta}}\,\mathrm{e}^{-x/2}\,\mathrm{d}x = 2\sqrt{\delta(1-\delta)}\,.$$

It follows that the stability condition for the BSC is given by

$$\lambda'(0)\rho'(1) < \frac{1}{2\sqrt{\delta(1-\delta)}}\,.$$

□

Example 8. [AWGC] For the AWGNC (see Example 3) we have

$$\mathrm{e}^{-r} = 2\int_0^\infty \sqrt{\frac{\sigma^2}{8\pi}}\,\mathrm{e}^{-\frac{(x-\frac{2}{\sigma^2})\sigma^2}{8}}\,\mathrm{e}^{-x/2}\,\mathrm{d}x = \mathrm{e}^{-\frac{1}{2\sigma^2}}\,.$$

Thus, the stability condition reduces to

$$\lambda'(0)\rho'(1) < \mathrm{e}^{\frac{1}{2\sigma^2}}\,.$$

□

Example 9. [Laplace channel] For the Laplace channel (see Example 4) we have

$$\mathrm{e}^{-r} = 2\int_0^{\frac{2}{\lambda}^+} [\frac{1}{4}\,\mathrm{e}^{-\frac{1}{\lambda}}\,\mathrm{e}^{x} + \frac{1}{2}\Delta_{\frac{2}{\lambda}}]\,\mathrm{e}^{-x/2}\,\mathrm{d}x = \mathrm{e}^{-\frac{1}{\lambda}}\frac{\lambda+1}{\lambda}\,.$$

The stability condition for the Laplace channel can therefore be written as

$$\lambda'(0)\rho'(1) < \mathrm{e}^{\frac{1}{\lambda}}\frac{\lambda}{1+\lambda}\,.$$

□

We conjecture that the converse of the condition above is sufficient to guarantee local convergence of the probability of error to zero.

Conjecture 1. *If $\lambda'(0)\rho'(1) < \mathrm{e}^{r}$, then there exists $\epsilon > 0$ such that if density evolution is initialized with a consistent message distribution P satisfying $\mathrm{Pr}_{err}(P) < \epsilon$, then the probability of error will converge to zero under density evolution.*

We remark that it suffices to prove the conjecture for $P = g_z, z < \epsilon$, where g_z was defined in (6).

To support the conjecture we have only been able to prove various partial results. Empirically, all our numerically optimized degree sequences fulfill this condition. We conjecture that degree sequences which approach capacity have to fulfill the said condition almost with equality.

3 Results

3.1 Good Degree Sequences

Using numerical optimization techniques described in more detail in Section 4, we searched for good degree sequences of rate one-half for various upper bounds on the maximum left degree d_l.

d_l	4	5	6	7	8	9	10	11	12
λ_2^*	0.38364	0.34648	0.34043	0.31480	0.30166	0.28321	0.27165	0.26269	0.25522
λ_2	0.38354	0.32660	0.33241	0.32395	0.30013	0.27684	0.25105	0.23882	0.24426
λ_3	0.04237	0.11960	0.24632	0.25094	0.28395	0.28342	0.30938	0.29515	0.25907
λ_4	0.57409	0.18393	0.11014				0.00104	0.03261	0.01054
λ_5		0.36988							0.05510
λ_6			0.31112						
λ_7				0.42511					
λ_8					0.41592				0.01455
λ_9						0.43974			
λ_{10}							0.43853		0.01275
λ_{11}								0.43342	
λ_{12}									0.40373
ρ_5	0.24123								
ρ_6	0.75877	0.78555	0.76611	0.43141	0.22919	0.01568			
ρ_7		0.21445	0.23389	0.56859	0.77081	0.85244	0.63676	0.43011	0.25475
ρ_8						0.13188	0.36324	0.56989	0.73438
ρ_9									0.01087
σ^*	0.9114	0.9194	0.9304	0.9438	0.9497	0.9540	0.9558	0.9572	0.9580
$(\frac{E_b}{N_0})^*_{dB}$	0.7820	0.7299	0.6266	0.5024	0.4483	0.4090	0.3927	0.3799	0.3727
p^*	0.1369	0.1384	0.1412	0.1447	0.1462	0.1473	0.1477	0.1481	0.1483

Table 1: Good degree sequences of rate one-half for the AWGNC and with maximal left degrees $d_l = 4, 5, 6, 7, 8, 9, 10, 11$ and 12. For each sequence the threshold value σ^*, the corresponding $(\frac{E_b}{N_0})^*$ in dB and $p^* = Q(1/\sigma^*)$, i.e., the input bit error probability of a hard-decision decoder are given. Also listed is λ_2^*, the maximal stable value of λ_2 for the given $\rho'(1)$ and for $\sigma = \sigma^*$.

The result of our search for the AWGNC is summarized in Tables 1 and 2. Table 1 contains those sequences with $d_l = 4, \cdots, 12$, whereas Table 2 contains sequences with $d_l = 15, 20, 30$ and 50. In each table columns correspond to one particular degree sequence. For each sequence the coefficients of λ and ρ are given as well as the threshold σ^*, the corresponding parameter $(E_b/N_0)^*$ in dB, and finally the raw bit error probability p^* of the input if it were quantized to 1 bit, i.e., $p^* = Q(1/\sigma^*)$, where $Q(x) := (1/\sqrt{2\pi})\int_x^\infty \mathrm{e}^{-y^2/2}\mathrm{d}y$ is (up to scaling) the complementary error function of Gaussian statistics.[8] Also listed is the maximal stable coefficient λ_2^* for the given $\rho'(1)$ and $\sigma = \sigma^*$, see Section 2.3. As can be seen, for every degree sequence $\lambda_2 < \lambda_2^*$, and the two values are fairly close.

The results are quite encouraging. Compared to regular LDPCCs for which the highest achievable threshold for the AWGNC is $\sigma^* = 0.88$, irregular LDPCCs have substantially higher thresholds.

[8] More precisely, $Q(x) := \frac{1}{2}\mathrm{erfc}(x/\sqrt{2})$.

d_l	15	20	30	50
λ_2^*	0.24446	0.23261	0.21306	0.18379
λ_2	0.23802	0.21991	0.19606	0.17120
λ_3	0.20997	0.23328	0.24039	0.21053
λ_4	0.03492	0.02058		0.00273
λ_5	0.12015			
λ_6		0.08543	0.00228	
λ_7	0.01587	0.06540	0.05516	0.00009
λ_8		0.04767	0.16602	0.15269
λ_9		0.01912	0.04088	0.09227
λ_{10}			0.01064	0.02802
λ_{14}	0.00480			
λ_{15}	0.37627			0.01206
λ_{19}		0.08064		
λ_{20}		0.22798		
λ_{28}			0.00221	
λ_{30}			0.28636	0.07212
λ_{50}				0.25830
ρ_8	0.98013	0.64854	0.00749	
ρ_9	0.01987	0.34747	0.99101	0.33620
ρ_{10}		0.00399	0.00150	0.08883
ρ_{11}				0.57497
σ^*	0.9622	0.9649	0.9690	0.9718
$(\frac{E_b}{N_0})^*_{dB}$	0.3347	0.3104	0.2735	0.2485
p^*	0.1493	0.1500	0.1510	0.1517

Table 2: Good degree sequences of rate one-half for the AWGNC and with maximal left degrees $d_l = 15, 20, 30$ and 50. For each sequence the threshold value σ^*, the corresponding $(\frac{E_b}{N_0})^*$ in dB and $p^* = Q(1/\sigma^*)$, i.e., the input bit error probability of a hard-decision decoder are given. Also listed is λ_2^*, the maximal stable value of λ_2 for the given $\rho'(1)$ and for $\sigma = \sigma^*$.

The threshold increases initially rapidly with d_l and for the largest investigated degree, $d_l = 50$, the threshold value is only 0.06dB away from capacity!

It is quite tempting to conjecture that the threshold will converge to the ultimate Shannon limit (which is equal to $\sigma_{\text{opt}} = 0.9787$ for rate one-half codes) as d_l tends to infinity. Unfortunately, as of this moment, we only have the presented empirical evidence to support this conjecture.

Although in this paper we focus mostly on the AWGNC and binary codes of rate one-half, the following examples show that the same techniques can be used to find very good degree sequences for other memoryless channels and different rates. We note that, for a particular rate, sequences which were optimized for the AWGNC are usually very good sequences for a large class of channels, including the Laplace channel and the BSC. Nevertheless, optimizing a degree sequence for a particular channel will generally give slightly better results.

Example 10. [AWGNC] In this example we consider codes of rate $r = \frac{8}{9}$ for the AWGNC channel. In this case, the Shannon bound for σ is $\sigma_{\text{opt}} = 0.528936$. Our optimizer produced the following degree distribution which gives a theoretical threshold of $\sigma^* = 0.5183$, $\lambda(x) := 0.1575x + 0.3429x^2 + 0.0363x^5 + 0.0590x^6 + 0.2790x^8 + 0.1253x^9$ and $\rho(x) := 0.8266x^{34} + 0.1345x^{35} + 0.0087x^{70} + 0.0302x^{71}$.

$\square$

Example 11. [BSC; $r = \frac{1}{2}$] The ultimate threshold for the BSC and rate one-half is $\delta_{\text{opt}} = 0.110028$. The best sequence we have found so far has $\delta^* = 0.106$ and is given by $\lambda(x) := 0.157581x + 0.164953x^2 + 0.0224291x^3 + 0.045541x^4 + 0.0114545x^5 + 0.0999096x^6 + 0.0160667x^7 + 0.00258277x^8 + 0.00454797x^9 + 0.000928767x^{10} + 0.0188361x^{11} + 0.0648277x^{12} + 0.0206867x^{13} + 0.000780516x^{14} + 0.0383603x^{15} + 0.0419398x^{16} + 0.0023117x^{19} + 0.00184157x^{20} + 0.0114194x^{22} + 0.0116636x^{28} + 0.0850183x^{39} + 0.01048x^{40} + 0.0169308x^{55} + 0.0255644x^{56} + 0.0364086x^{70} + 0.0869359x^{74}$ and $\rho(x) := 0.25x^9 + 0.75x^{10}$.

$\square$

Example 12. [Laplace channel; $r = \frac{1}{2}$] The ultimate threshold for the binary-input Laplace channel of rate $\frac{1}{2}$ (as given by the Shannon formula) is equal to $\lambda_{\text{opt}} = 0.752$. We found the following degree sequence which has a threshold above $\lambda^* = 0.74$, $\lambda(x) := 0.0869518x + 0.26831x^2 + 0.0928357x^3 + 0.0214918x^4 + 0.0120479x^5 + 0.00416287x^6 + 0.010689x^7 + 0.00271656x^8 + 0.00478356x^9 + 0.000976879x^{10} + 0.0198119x^{11} + 0.0681859x^{12} + 0.0229726x^{13} + 0.0782138x^{14} +$

$0.00315161x^{15} + 0.0033345x^{16} + 0.00251733x^{17} + 0.00243145x^{19} + 0.00224413x^{20} + 0.00372345x^{21} + 0.012011x^{22} + 0.00763323x^{24} + 0.0615922x^{25} + 0.0122678x^{28} + 0.00113751x^{53} + 0.00565326x^{54} + 0.0178079x^{55} + 0.0268887x^{56} + 0.000395817x^{59} + 0.00229811x^{64} + 0.0144568x^{73} + 0.126305x^{74}$ and $\rho(x) := 0.25x^9 + 0.75x^{10}$.

$\square$

3.2 Simulation Results

The concentration results proved in [1] guarantee that for sufficiently large block lengths almost every code in the given ensemble will have vanishing probability of error for channels with parameters below the calculated threshold. Nevertheless, the lengths required by the proofs are far beyond any practical values and one might expect that medium sized codes will deviate significantly from the predicted performance. For regular LDPCC's, it was demonstrated in [1] that the actual convergence is much faster and that already realistically sized block codes perform close to their predicted value.

For irregular codes finite length effects not only include the deviation of the input variance from its mean and a non-zero probability of small loops but also the deviation of a given support tree from its average, i.e., for a given node the fraction of neighbors of this node with a given degree might deviate from its expected value. This effect is expected to influence the finite length performance more severely for larger d_l and d_r. Further, when operating very close to a degree sequence's threshold it will require a large number of iterations (one thousand or more) to reach rest error probabilities of, say, 10^{-5}. (In the limit the number of iterations converges to ∞. Typically, however, a small margin from a sequence's threshold is enough to drastically reduce the required number of iterations.) Given that the number of loop-free iterations only grows logarithmically in the block length it seems a priori doubtful that simulation results can closely match the predicted limit for practical lengths.

Fortunately, the actual convergence of the performance of finite length codes to the asymptotic performance is much faster than predicted this way. Fig. 3 shows the performance of particular LDPCCs. The chosen lengths start at one thousand and go until one million. More precisely, the lengths presented are $10^3, 10^4, 10^5$, and 10^6. The maximum degrees appearing on the left are $9, 20, 50$, and 50 respectively. For length 10^3 the error rates are given for systematic bits. (A specific encoder was constructed.) These graphs were not chosen entirely randomly. The degree

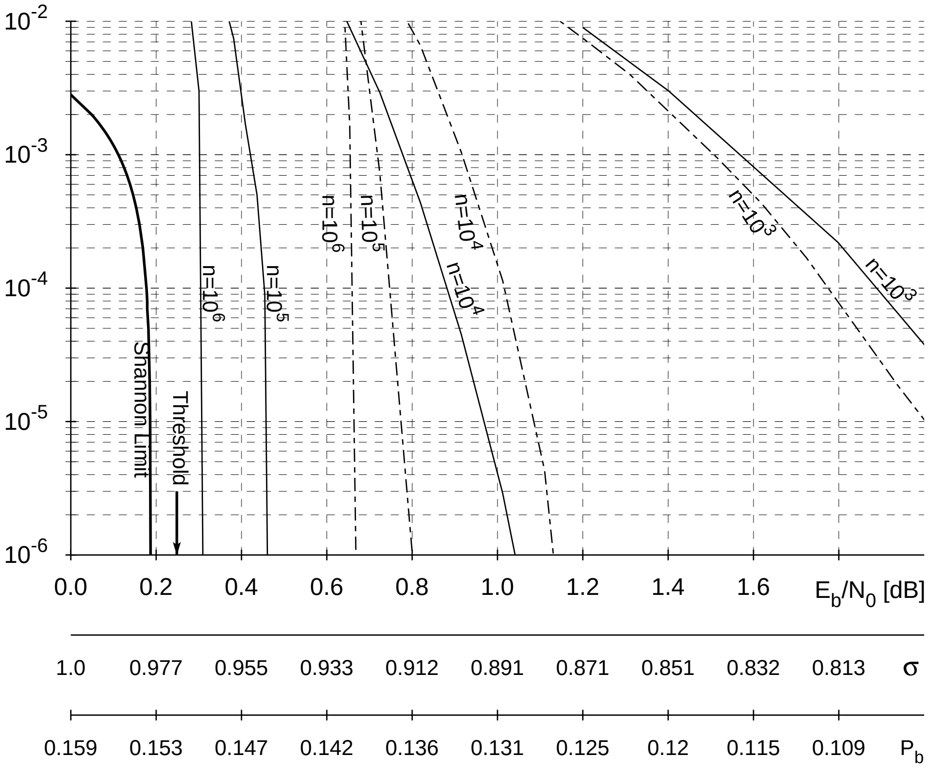

Figure 3: Comparison between turbo codes (dashed curves) and LDPCCs (solid curves) of lengths $n = 10^3, 10^4, 10^5$ and 10^6. All codes are of rate one-half. Observe that longer LDPCCs outperform turbo codes and that the gap becomes the more significant with larger n. For short lengths it appears that the structure in turbo-codes gives them an edge over LDPCC codes despite having a lower threshold.

two nodes were put in loop free and all of them correspond to non-systematic bits. Some effort was also made to reduce the number of small loops in these graphs. For length 10^4 and above the error rate was given over all of the bits in the codeword. The length 10^6 graph is random except that double edges and loops with two variable nodes were avoided. For length 10^4 and 10^5 the degree two nodes were put in loop free and some small loop removal was performed. We note that, particularly for small lengths, better performance can be achieved by using degree sequences with smaller values of d_l even though such sequences have a smaller threshold.[9] We see that for one million bits the actual performance is quite close to the predicted performance and that also for a length of 10^5 bits, this code handily beats the best known turbo code of the same length.

[9]The sequences presented here are those giving the highest threshold under a maximum degree constraint. For small graphs it is not always best to pick the sequence with the highest threshold. When designing sequences for small graphs, it may be advantageous to look for the highest possible threshold under an appropriate constraint on the allowed number of iterations.

At one million bits the code is less than 0.13dB away from capacity at bit error probabilities of 10^{-6}. Given that LDPCCs have slightly lower complexity, are fully parallelizable, and allow many different decoding algorithms with a far ranging trade-off between performance and complexity, LDPCCs can be considered serious competitors to turbo-codes.

Although we spent a considerable amount of computing time on the optimization it is clear that any given sequence can be (slightly) improved given enough patience.

4 Optimization

In this section we briefly describe the optimization techniques that we used to obtain degree sequences with large thresholds.

The following general remarks apply to any numerical optimization technique. First, formally, the threshold is defined as the supremum of all channel parameters for which the probability of error under density evolution converges to zero. By Corollary 1 this is equivalent to requiring that the message distribution converge to Δ_∞. In practice, at best we can verify that the probability of error reaches a value below a prescribed ϵ. One may then wonder if, by choosing ϵ small enough, this automatically implies the convergence to zero probability of error.

As discussed in more detail in Section 2.3, the stability condition is a first step in this direction. Unfortunately, we have only been able to prove necessity of the stability condition. Conjecture 1 expresses our belief that the same conditions on the degree sequence are indeed sufficient to guarantee convergence. In practice the issue of convergence is not of great concern since we always allow a finite (but small) probability of error.

Secondly, in order to perform the computations we need to quantize the involved quantities. This quantization leads to a quantization error and this error might accumulate over the course of many iterations, rendering the computations useless. This problem can be circumvented in the following way. By carefully performing the quantization one can assure that the quantized density evolution corresponds to the density evolution of a quantized message passing scheme. Since belief propagation is optimal, such a quantized version is sub-optimal and, hence, the reported thresholds can be thought of as lower bounds on the actual thresholds.

4.1 Local Optimization

To find good degree sequences we started with the following simple hill climbing approach. Fix a small rest error probability ϵ and a maximum number of iterations m. Start with a given degree sequence and determine the maximum *admissible* channel parameter, i.e., the maximum channel parameter such that the error probability after m iterations is below ϵ. Now apply a small change to the degree sequence and check if it has either a larger admissible channel parameter or at least a smaller rest error probability after m iterations. If so, declare the new sequence to be your currently best sequence, otherwise keep the original sequence. The same basic step is then repeated a large number of times.

The search for good sequence can be substantially accelerated by appropriately limiting the search space. We found, for example, that very good sequences exist with only a few non-zero terms. In particular, it suffices to allow two or three non-zero degrees on the right (and these degrees can be chosen consecutively) and to limit the non-zero degrees on the left to the degrees 2, 3, the maximum left degree d_l, and, possibly, a few well-chosen degrees in-between.

A further substantial savings in running time can be achieved as follows. For a particular degree sequence (λ, ρ), Fig. 4 shows the evolution of the bit error probability as a function of the iteration number. An even clearer picture evolves if we plot the *decrease* of the bit error probability as a function of the current bit error probability p_b. This is shown in Fig. 5.

As can be seen in these figures, after an initial swift decrease in the bit error probability the procedure almost comes to a halt at $p_b = 0.111$ with decreases in bit error probability of only 2.6×10^{-5} per iteration. The convergence then speeds up again until it hits another low at $p_b = 0.056$ and then later again at $p_b = 0.022$. At these three *critical points* the outgoing message distribution is *almost* a fixed point of the equation system corresponding to one iteration. Indeed, if the parameter σ were slightly increased then the iteration could not overcome any of those points and one can verify that there arise corresponding fixed points of density evolution.

Provided that the fixed points are stable, the message distributions at these points are *continuous* functions of the degree sequence. Hence, a small change in the degree sequence causes only small changes in the associated fixed point distributions. Furthermore, if the fixed points are stable, then this affords a certain memorylessness to the density evolution because they serve as local attractors. Small perturbations to the path will not matter once the domain of convergence of the

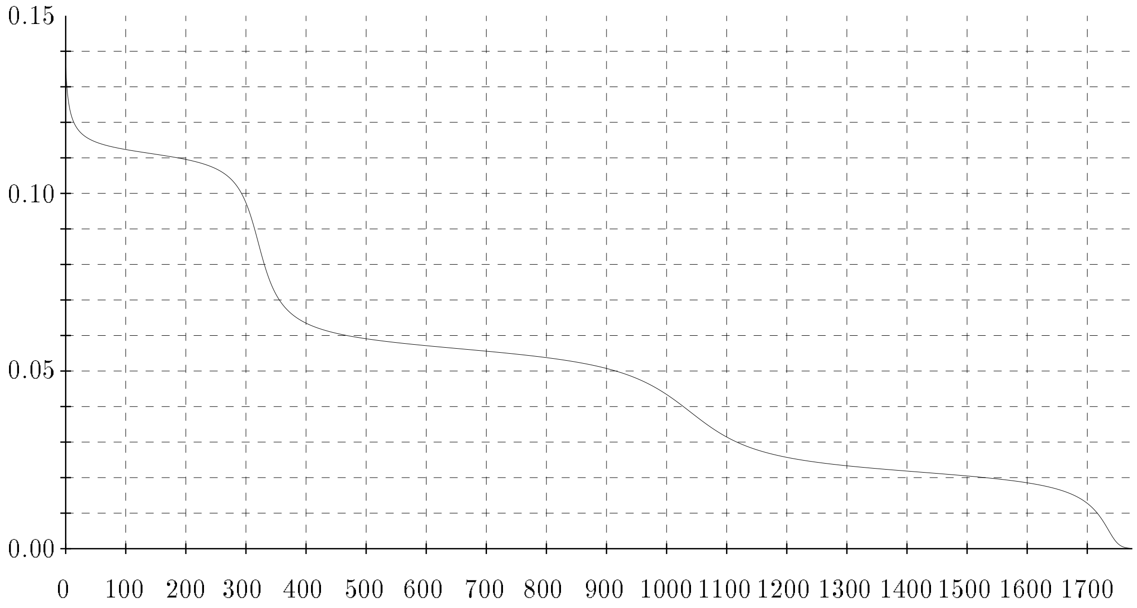

Figure 4: Evolution of the bit error probability as a function of the iteration number.

fixed point is entered and, once the fixed point is found, the path that leads to it is irrelevant. In practice we observe that the points at which the density evolution gets stuck are indeed stable fixed points. The fixed point theorem in Section 2.2 shows that fixed points which are limits of density evolution must be at least marginally stable. The above considerations suggests the following scheme. Assume we determined the critical points (near fixed points, or likely fixed points for a slightly worse initial distribution) for a particular degree sequence and we would like to determine the merit of a particular small change of the degree sequence. Rather than starting with the initial distribution and then checking if (and how fast) this initial distribution converges to a delta at infinity, one can memorize the distributions at the critical points of the original degree sequence and then determine how the proposed change affects the speed of convergence locally at these points. Once a promising change has been found, the merit of this change can be verified by starting with the initial degree sequence. Typically only a few iterations are necessary at each critical point to determine if the change in degree sequence did improve the convergence or not. This has to be compared to hundreds of iterations or even thousands of iterations which are necessary if one starts with the initial distribution.

In the last scheme we made use of the distributions at the "critical points" to find promising changes of the degree sequences. The following schemes extend this idea even further, and the resulting

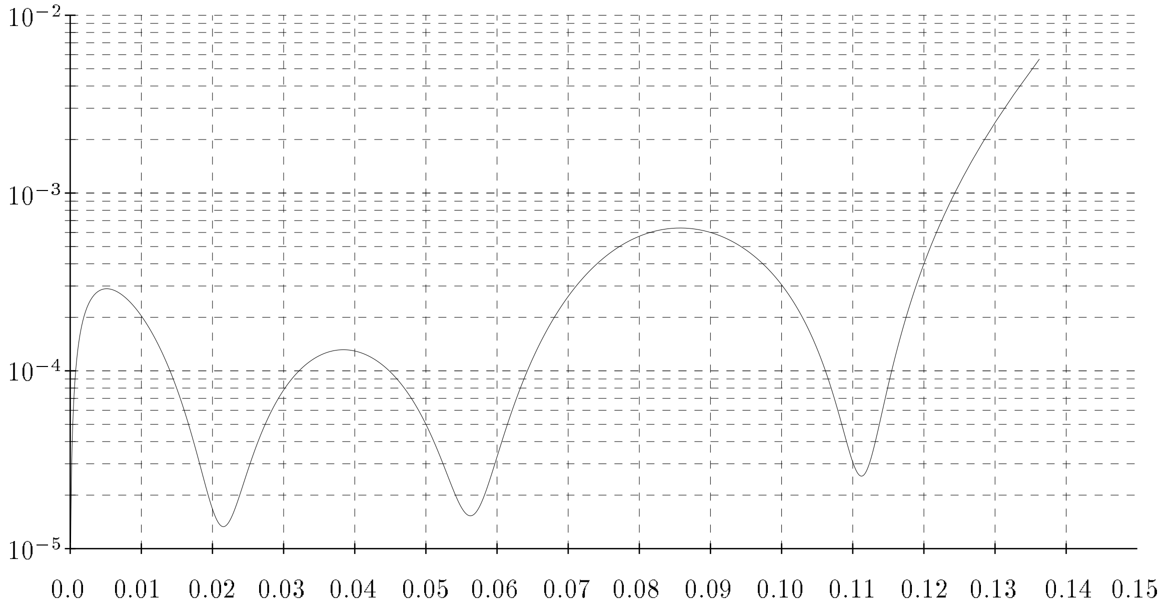

Figure 5: The decrease of the bit error probability as a function of the current bit error probability.

algorithms are reminiscent of the algorithms used in the erasure channel case. For simplicity we will only describe the optimization of the left degree sequence. The extension to the right degree sequence and to joint optimization should be quite apparent.

Assume we are given a degree sequence (λ, ρ), a particular channel parameter σ, and a target probability of error ϵ. Let $\{p_\ell\}_{\ell=0}^m$ be the sequence of error probabilities of the belief propagation algorithm. More precisely, p_0 is the initial error probability, p_ℓ is the probability of error after the ℓ-th iteration, and $p_m \leq \epsilon < p_{m-1}$. Assume we want to find a new degree sequence $\tilde{\lambda}$ which achieves the target probability of error in fewer iterations or achieves a lower target in the same number of iterations.

Define the matrix $A_{\ell,j}$, $1 \leq \ell \leq m$, $2 \leq j \leq d_r$. The entry $A_{\ell,j}$ is the error probability which results if we run the belief propagation decoder for $(\ell - 1)$ steps assuming that the left degree sequence is λ followed by one step in which we assume that the left degree sequence is a singleton with all its mass on the degree j. Note that the actual error probability after the ℓ-th iteration, p_ℓ, can be expressed in terms of $A_{\ell,j}$ as

$$p_\ell = \sum_{j=2}^{d_l} A_{\ell,j} \lambda_j \ .$$

Let us define a function $p(t)$ for $t \in [0,m]$ by linearly interpolating the p_ℓ, setting $p(\ell) = p_\ell$. Define

$$L(\lambda) := \int_{p_m}^{p_0} \left(-\frac{\mathrm{d}p}{\mathrm{d}t}(x) \right)^{-1} \mathrm{d}x.$$

We interpret L as the number of iterations required to take the initial probability of error p_0 down to p_m. Using the expression above, we can write down the gradient of $L(\lambda)$ with respect to λ. In particular, for a perturbation h we can compute $D_h L(\lambda) := \frac{d}{d\eta} L(\lambda + \eta h)\mid_{\eta=0}$ as

$$D_h L = \int_{p_m}^{p_0} \left(\frac{\mathrm{d}p}{\mathrm{d}t}(x) \right)^{-2} D_h \left(\frac{\mathrm{d}p}{\mathrm{d}t}(x) \right) \mathrm{d}x.$$

Returning to the discrete representation this is equivalent to

$$D_h L = \sum_{j=2}^{d_l} h_j \left(\sum_{\ell=1}^{m} \frac{A_{\ell,j} - p_\ell}{p_{\ell-1} - p_\ell} \right).$$

Thus, we observe that the gradient of $L(\lambda)$ is given by

$$\frac{\mathrm{d}}{\mathrm{d}\lambda_j} L(\lambda) = \sum_{\ell=1}^{m} \frac{A_{\ell,j} - p_\ell}{p_{\ell-1} - p_\ell}.$$

There are two ways we can exploit this expression. One is to use the (negative) gradient direction to do hill climbing, and the other is to globally optimize the linearized approximation of L. In either case we must incorporate the constraints on λ.

Let $\tilde{\lambda}$ be an alternative degree sequence. Clearly, $\tilde{\lambda}$ has to be a probability mass function, i.e.,

$$\sum_{j=2}^{d_l} \tilde{\lambda}_j = 1, \tag{11}$$

and further it has to correspond to a code of equal rate, i.e.,

$$\sum_{j=2}^{d_l} \frac{\tilde{\lambda}_j}{j} = \sum_{j=2}^{d_l} \frac{\lambda_j}{j}. \tag{12}$$

Let h be the negative gradient direction of L. If we set $\tilde{\lambda} = \lambda + \eta h$ (for positive η) then the above constraints may not be satisfied. However, among sequences satisfying the constraints the one closest to $\lambda + \eta h$ in Euclidean distance can be easily computed by alternating projections. Two projections are required: the first is orthogonal projection of h onto the subspace determined by $\sum_j h_j = 0$ (total probability constraint) and $\sum_j \frac{1}{j} h_j = 0$ (rate constraint), and the second projection sets $\eta h_j = -\lambda_j$ if, prior to the projection, $\eta h_j + \lambda_j < 0$. Note that an alternative interpretation is to project the gradient direction h onto the convex polytope of admissible directions. One can then compute the maximum step size η for which the constraints remain satisfied and then recompute the projection at that point. In this way one can easily walk along the projected gradient direction to look for an improved degree sequence.

Let us now consider the second way to exploit the gradient expression for L. Let $\tilde{p}_\ell := \sum_{j=2}^{d_l} A_{\ell,j}\tilde{\lambda}_j$. Then we have

$$L(\tilde{\lambda}) \simeq \sum_{\ell=1}^{m} \frac{p_{\ell-1} - \tilde{p}_\ell}{p_{\ell-1} - p_\ell}. \tag{13}$$

This approximation is valid as long as $\tilde{\lambda}$ does not differ too much from λ, i.e., assuming that the message distributions corresponding to λ and $\tilde{\lambda}$ are not too different, if

$$\max_{\ell} \frac{|p_\ell - \tilde{p}_\ell|}{p_{\ell-1} - p_\ell} < \delta, \tag{14}$$

where $\delta << 1$, and if

$$\tilde{p}_\ell < p_{\ell-1},\ 1 \leq \ell \leq m. \tag{15}$$

Recall that we want to minimize $L(\tilde{\lambda})$. Since the right hand side of (13) is (up to a constant) a linear function in the degree sequence and since the constraints stated in (11), (12), (14) and (15) are also linear, this can be (approximately) accomplished by means of a linear program. The same procedure is then applied repeatedly in an attempt to converge to a good degree sequence.

Since both approaches are local optimizations it is appropriate to repeat the optimization with various initial conditions.

4.2 Global Optimization: Differential Evolution

The code design problem as described above belongs to the class of nonlinear constraint satisfaction problems with continuous space parameters, a problem class where Differential Evolution (DE) [13] has proven to be very effective [14, 15]. This strategy has been successfully applied to the design of good erasure codes obtained from bipartite graphs, as described in [7].

Differential evolution is a robust optimizer for multivariate functions. Starting from an initial random population of points (interpreted as vectors), the procedure iteratively updates the population using two basic operations: *mutation* and *recombination*. Under mutation differences of random population vectors are formed, scaled, and added to certain population members. Scaling as well as the mechanism of population member selection is controlled at run-time by input variables. The operation of recombination is a 'genetic' part of the algorithm. Here two population members are chosen at random, split at some index according to a given probability distribution on the set of indices, and the corresponding parts of these vectors are recombined. After these operations are performed, the function is evaluated at the original population member and the new member, and the one with the highest cost function value is chosen to belong to the new population. The procedure is repeated for a prescribed number of generations and at each round the population member with the highest cost function value is recorded.

We experimented with DE by using different cost functions and different population characteristics. In the following, we will only discuss the approaches that produced the best results.

Our cost function maximized the value of the error-threshold: for the BSC and the Gaussian channel, the initial error distribution is a function of a single parameter. The goal of the optimization was maximizing this parameter subject to the condition that the iteration process finishes successfully.

Since DE is a continuous optimizer, we found it convenient to let the parameter space be a continuous space as well. For this, we introduced *fractional phantom distributions*: we let the polynomials λ and ρ take on the general form $\sum_i \lambda_i x^{i-1}$ (similarly for ρ), where now both the λ_i *and the degree* i could take any positive real value. The real degree distribution is obtained from this phantom distribution as $\sum_i (\lambda_{i1} x^{\lfloor i \rfloor -1} + \lambda_{i2} x^{\lceil i \rceil -1})$, where λ_{i1} and λ_{i2} are uniquely determined via the equations

$$\lambda_{i1} + \lambda_{i2} = \lambda_i, \quad \text{and} \quad \frac{\lambda_{i1}}{\lfloor i \rfloor} + \frac{\lambda_{i2}}{\lceil i \rceil} = \frac{\lambda_i}{i}.$$

This way we are guaranteed to obtain a degree distribution which respects the rate-constraints for the code.

By allowing fractional degrees, we could in effect force the program to choose (close to) optimal degrees. This resulted in a major reduction of the dimensionality of the parameter space, hence the running time, and in the sparsity of the sequences obtained.

References

[1] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding." submitted IEEE IT.

[2] D. Spielman, "Linear-time encodeable and decodable error-correcting codes," *IEEE Trans. on Information Theory*, vol. 42, November 1996.

[3] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 249–258, 1998.

[4] T. Richardson, "Efficient encoding of Low Density Parity Check Codes." in preparation.

[5] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proceedings of the* 29*th annual ACM Symposium on Theory of Computing*, pp. 150–159, 1997.

[6] M. Luby, M. Mitzenmacher, and A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," in *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 364–373, 1998.

[7] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution." To appear in Proceedings of GECCO'99.

[8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proceedings of ICC'93*, (Geneve, Switzerland), pp. 1064–1070, May 1993.

[9] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, Massachusetts: M.I.T. Press, 1963.

[10] R. Wheeden and A. Zygmund, *Measure and Integral.* Marcel Dekker, Inc., 1977.

[11] A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity." Submitted to AAECC-13, 1999.

[12] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis.* Chapman & Hall, 1995.

[13] K. Price and R. Storn, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[14] R. Storn, "Differential evolution design of an IIR-filter with requirements of magnitude and group delay," in *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 268–273, 1996.

[15] R. Storn, "On the usage of differential evolution for function optimization," in *NAFIPS'96*, pp. 519–523, 1996.