LP Decoding of Regular LDPC Codes in Memoryless Channels

Nissim Halabi ∗ Guy Even †

March 26, 2010

Abstract

We study error bounds for linear programming decoding of regular LDPC codes with logarithmic girth. For memoryless binary-input AWGN channels, we prove thresholds for exponentially small decoding errors. Specifically, for (3, 6)-regular LDPC codes, we prove thresholds of $\sigma = 0.735$ provided that the girth is greater than 88 and $\sigma = 0.605$ for girth at least 4. For memoryless binary-input output-symmetric (MBIOS) channels, we prove exponentially small error bounds.

Our proof is an extension of a recent paper of Arora, Daskalakis, and Steurer [STOC 2009] who presented a novel probabilistic analysis of LP decoding over a binary symmetric channel. Their analysis is based on the primal LP representation and has an explicit connection to message passing algorithms. We extend this analysis to any MBIOS channel.

Our results improve the threshold for the AWGN channel presented by Koetter and Vontobel [ITAW 2006]. They proved a threshold of $\sigma = 0.5574$ for LP decoding of (3, 6)-regular codes working with the dual LP for girth at least 4.

∗School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail:nissimb@eng.tau.ac.il.
†School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail:guy@eng.tau.ac.il.
1 Introduction

Low-density parity-check (LDPC) codes were invented by Gallager [Gal63] in 1963. Gallager also invented the first type of message-passing iterative decoding algorithm, known today as the sum-product algorithm for a-posteriori probability (APP) decoding. Till the 1990s, iterative decoding systems were forgotten with a few exceptions such as the landmark paper of Tanner [Tan81] in 1981, who founded the study of codes defined by graphs. LDPC codes were rediscovered [MN96] after the discovery of turbo-codes [BGT93]. LDPC codes have attracted a lot of research since empirical studies demonstrate excellent decoding performance using iterative decoding methods. Among the main results is the density-evolution technique for analyzing and designing asymptotic LDPC codes [RU01]. A density-evolution analysis computes a threshold for the noise. This means that if the noise in the channel is below that threshold, then the decoding error diminishes exponentially as a function of the block length. The threshold results of [RU01] hold for a random code from an ensemble of LDPC codes.

Feldman et al. [Fel03, FWK05] suggested a decoding algorithm for linear codes that is based on linear programming. Initially, this idea seems to be counter-intuitive since codes are over $GF(2)^n$, whereas linear programming is over $\mathbb{R}^n$. Following ideas from approximation algorithms, linear programming (LP) is regarded as a fractional relaxation of an integer program that models the problem of decoding. One can distinguish between integral solutions (vertices) and non-integral vertices of the LP. The integral vertices correspond to codewords, whereas the non-integral vertices are not codewords and are thus called pseudocodewords. This algorithm, called LP-decoding, has two main advantages: (i) it runs in polynomial time, and (ii) when successful, LP-decoding provides an ML-certificate, i.e., a proof that its outcome agrees with maximum-likelihood (ML) decoding.

Koetter and Vontobel showed that LP decoding is equivalent to graph cover decoding [VK05]. Abstractly, graph cover decoding proceeds as follows. Given a received word, graph cover decoding considers all possible $M$-covers of the Tanner graph of the code (for every integer $M$). For every $M$-cover graph, the variables are assigned $M$ copies of the received word.
Maximum-likelihood (ML) decoding is applied to obtain a codeword in the code corresponding to the $M$-cover graph. The “best” ML-decoding result is selected among all covers. This lifted codeword is then projected (via averaging) to the base Tanner graph. Obviously, this averaging might yield a non-integral solution, namely, a pseudocodeword as in the case of LP decoding. Although this description of graph cover decoding is not constructive, it provides combinatorial insight for pseudocodewords.

LP decoding has been applied to several codes, among them: RA codes, turbo-like codes, LDPC codes, and expander codes. Decoding failures have been characterized, and these characterizations enabled proving word error bounds for RA codes, LDPC codes, and expander codes (see e.g., [FK04, HE05, KV06, FS05, FMS$^+$07, DDKW08, ADS09]). Experiments indicate that message-passing decoding is likely to fail if LP-decoding fails [Fel03, VK05].

1.1 Previous Results

Feldman et al. [FMS$^+$07] were the first to show that LP decoding corrects a constant fraction of errors for expander codes over an adversarial bit flipping channel. For example, for a specific family of rate $\frac{1}{2}$ LDPC expander codes, the proved that LP decoding can correct $0.000175n$ errors. This kind of analysis is worst-case in its nature, and the implied results are quite far from the performance of LDPC codes observed in practice over binary symmetric channels (BSC). Daskalakis et al. [DDKW08] initiated an average-case analysis of LP decoding for LDPC codes over a probabilistic bit flipping channels. For a certain family of LDPC expander codes over a binary symmetric channel (BSC) with bit flipping probability $p$, they proved that LP decoding recovers the transmitted codeword with high probability up to noise threshold of $p = 0.002$. This proved threshold for LP decoding is rather weak compared to thresholds proved for belief propagation (BP) decoding over the BSC. For example, even for $(3, 6)$-regular LDPC codes, the BP threshold is $p = 0.084$, and one would expect LDPC expander codes to be much better. Both of the results in [FMS$^+$07] and [DDKW08] were proved by analysis of the dual LP solution based on expansion arguments. Moreover, the analysis was shown only for the BSC, and it is not clear whether it can be applied to other probabilistic channels.
Koetter and Vontobel [KV06] analyzed LP-decoding of regular LDPC codes using girth arguments and the dual LP solution. They proved the first threshold for LP-decoding of LDPC codes over memoryless channels (other than the BSC) for which the decoding error decreases exponentially. They prove thresholds that depend only on the degree of the variable nodes (and not on the codelength or the girth). The decoding errors for noise below the threshold decrease doubly-exponential in the girth of the factor graph (for any memoryless symmetric channel). When applied to LP-decoding of \((3,6)\)-regular LDPC codes over BSC with crossover probability \(p\), they achieved a threshold \(p = 0.01\). For binary-input additive white Gaussian noise channel with noise variance \(\sigma^2\) (BI-AWGN(\(\sigma\))), they achieved a threshold \(\sigma = 0.5574\). The question of closing the gap to \(\sigma = 0.881\), which is the BP threshold for the same family of codes over a BI-AWGN(\(\sigma\)), remains open.

Recently, Arora et al. [ADS09] presented a novel probabilistic analysis of the primal solution of LP-decoding for regular LDPC codes over a BSC using girth arguments. They proved error bounds that are inverse doubly-exponential in the girth and thresholds much closer to the performance of BP decoding. Their technique is based on a weighted decomposition of every codeword to a finite set of structured trees. They proved a sufficient condition, called local-optimality, for the optimality of a decoded codeword based on this decomposition. They use a min-sum process on trees to bound the probability that the local-optimality holds. A probabilistic analysis of the min-sum process is applied to the structured trees of the decomposition, and yields error bounds for LP-decoding.

1.2 Our Results

In this work, we extend the analysis in [ADS09] from BSC to any memoryless binary-input output-symmetric (MBIOS) channel. We prove bounds for exponentially small error probability of LP-decoding for regular LDPC codes with logarithmic girth over MBIOS channels. We also prove thresholds for exponentially small decoding errors of LP-decoding for regular LDPC codes with logarithmic girth in memoryless binary-input AWGN channels. Specifically, we prove, for \((3,6)\)-regular LDPC codes with logarithmic girth, a threshold of \(\sigma = 0.605\) pro-
vided that the girth is at least 4, and \( \sigma = 0.735 \) for girth greater than 88, thus decreasing the gap to the BP asymptotic threshold.

In our analysis we utilize the combinatorial interpretation of LP-decoding via graph covers [VK05] to simplify some of the proofs in [ADS09]. Specifically, using the equivalence of graph cover decoding and LP-decoding in [VK05], we obtain a simpler proof that local-optimality suffices for LP optimality.

**Our Main Result:**

**Theorem 1.** Let \( G \) denote a \((d_L, d_R)\)-regular bipartite graph with girth \( \Omega(\log n) \), and let \( \mathcal{C}(G) \subset \{0, 1\}^n \) denote the low-density parity-check code defined by \( G \). Let \( x \in \mathcal{C}(G) \) be a codeword. Consider the BI-AWGNC(\( \sigma \)), and suppose that \( y \in \mathbb{R}^n \) is the word obtained from the channel given \( x \). Then, \( x \) is the unique optimal solution to the LP decoder with probability at least \( 1 - \exp(-n^\gamma) \) for some constant \( \gamma > 0 \), provided that one of the following conditions holds:

1. \((d_L, d_R) = (3, 6), \text{girth}(G) > 88, \text{and } \sigma \leq 0.735.\)
2. \((d_L, d_R) = (3, 6), \text{girth}(G) \geq 4, \text{and } \sigma \leq 0.605.\)

3. 

\[
\min_{t \geq 0} \left( (d_R-1)e^{-t} \int_{-\infty}^{\infty} (1-F_X(x))^{d_R-2} f_X(x)e^{-tx} dx \right) \cdot (d_R-1) e^{2t^2\sigma^2-1} \cdot \left( (d_R-1) e^{2t^2\sigma^2-1} \right)^{1/(d_L-2)} < 1,
\]

where \( f_X(\cdot) \) and \( F_X(\cdot) \) denote the p.d.f. and c.d.f. of a Gaussian random variable with zero mean and standard deviation \( \sigma \), respectively.

Theorem 1 generalizes to MBIOS channels as follows.

**Theorem 2.** Let \( G \) denote a \((d_L, d_R)\)-regular bipartite graph with girth \( \Omega(\log n) \), and let \( \mathcal{C}(G) \subset \{0, 1\}^n \) denote the low-density parity-check code defined by \( G \). Consider an MBIOS channel, and suppose that \( y \in \mathbb{R}^n \) is the word obtained from the channel given \( x = 0^n \). Let
\( \lambda \in \mathbb{R} \) denote the log-likelihood ratio of received channel observations, and let \( f_\lambda(\cdot) \) and \( F_\lambda(\cdot) \) denote the p.d.f. and c.d.f. of \( \lambda(y_i) \), respectively. Then, LP decoding succeeds with probability at least \( 1 - \exp(-n^\gamma) \) for some constant \( \gamma > 0 \), provided that

\[
\min_{t \geq 0} \left( (d_R - 1) \int_{-\infty}^{\infty} \left( 1 - F_\lambda(z) \right)^{d_R - 2} f_\lambda(z) e^{-tz} dz \right) \cdot \left( (d_R - 1) \mathbb{E} e^{-t\lambda} \right)^{1/(d_L - 2)} < 1.
\]

2 Preliminaries

Low-density parity-check codes and factor graph representation. A code \( C \) with block length \( n \) over \( \mathbb{F}_2 \) is a subset of \( \mathbb{F}_2^n \). Vectors in \( C \) are referred to as codewords. An \([n, k]\) binary linear code is a \( k \)-dimensional vector subspace of the vector space \( \mathbb{F}_2^n \). A parity-check matrix for an \([n, k]\) binary linear code \( C \) is an \((n - k) \times n\) matrix \( H \) whose rows span the space of vectors orthogonal to \( C \). That is, \( H \) is a full rank matrix such that \( C = \{ x \in \mathbb{F}_2^n : Hx = 0 \} \).

The factor graph representation of a code \( C \) is a bipartite graph \( G \) that represents the matrix \( H \). The factor graph \( G \) is over variable nodes \( V_L \triangleq [n] \) and check nodes \( V_R \triangleq [m] \), where \( m = n - k \). An edge \((i, j)\) connects variable node \( i \) and check node \( j \) if \( H_{j,i} = 1 \). The variable nodes correspond to bits of the codeword and the check nodes correspond to the rows of \( H \). Every bipartite graph defines a parity check matrix. If the bipartite graph is \((d_L, d_R)\)-regular\(^1\) for some constants \( d_L \) and \( d_R \), then it defines a \((d_L, d_R)\)-regular low-density parity-check (LDPC) code.

LP decoding over memoryless channels. Let \( X_i \in \{0, 1\} \) and \( Y_i \in \mathbb{R} \) denote random variables that correspond to the \( i \)th transmitted symbol (channel input) and the \( i \)th received symbol (channel output), respectively. A memoryless binary-input output-symmetric (MBIOS) channel is defined by a conditional probability \( \mathbb{P}_{Y_i/X_i}(y_i/x_i) \triangleq \mathbb{P}(Y_i = y_i/X_i = x_i) \). The log-likelihood ratio (LLR) vector \( \lambda \in \mathbb{R}^n \) for a received word \( y \in \mathbb{R}^n \) is defined by

\[
\lambda_i(y_i) \triangleq \ln \frac{\mathbb{P}_{Y_i/X_i}(y_i/0)}{\mathbb{P}_{Y_i/X_i}(y_i/1)},
\]

\(^1\)That is, a bipartite graph with left vertices of degree \( d_L \) and right vertices of degree \( d_R \).
for $i \in [n]$. For a linear code $C$, Maximum-Likelihood (ML) decoding is equivalent to

$$\hat{x}^{ML}(y) = \arg \min_{x \in \text{conv}(C)} \langle \lambda(y), x \rangle,$$

(1)

where $\text{conv}(C)$ denotes the convex hull of the set $C$.

Solving in general the optimization problem in Equation (1) for linear codes is intractable. Furthermore, the decision problem of ML decoding remains NP-hard even for the class of left-regular LDPC codes [XH07]. Feldman et al. [Fel03, FWK05] introduced a linear programming relaxation for the problem of ML decoding of linear codes. Given a factor graph $G$, for every $j \in V_R$, denote by $C_j$ the set of binary sequences that satisfy parity check constraint $j$,

$$C_j \triangleq \{ x \in \mathbb{F}_2^n : \sum_{i \in \mathcal{N}(j)} x_i = 0 \ (\text{mod}2) \}.$$

Let $\mathcal{P}(G) \triangleq \bigcap_{j \in V_R} \text{conv}(C_j)$ denote the fundamental polytope [Fel03, FWK05, VK05] of a factor graph $G$. For LDPC codes, the fundamental polytope is defined by a linear number of constraints. Given an LLR vector $\lambda$ for a received word $y$, LP decoding consists of solving the following optimization problem

$$\hat{x}^{LP}(y) \triangleq \arg \min_{x \in \mathcal{P}(G)} \langle \lambda(y), x \rangle,$$

(2)

which can be solved in time polynomial in $n$ using linear programming.

Let us denote by BI-AWGNC($\sigma$) the binary input additive white Gaussian noise channel with noise variance $\sigma^2$. The channel input $X_i$ at time $i$ is an element of $\{\pm 1\}$ since we map a bit $b \in \{0, 1\}$ to $(-1)^b$. Given $X_i$, the channel outputs $Y_i = X_i + \phi_i$ where $\phi_i \sim \mathcal{N}(0, \sigma^2)$. For BI-AWGNC($\sigma$), $\lambda_i(y_i) = \frac{2y_i}{\sigma^2}$. Note that the optimal ML and LP solutions are invariant under positive scaling of the LLR vector $\lambda$. 


3 On the Connections between Local Optimality, Global Optimality, and LP Optimality

Let $x \in C(G)$ denote a codeword and $\lambda(y) \in \mathbb{R}^n$ denote an LLR vector for a received word $y \in \mathbb{R}^n$. Following [ADS09], we consider two questions: (i) does $x$ equal $\hat{x}_{ML}(y)$? and (ii) does $x$ equal $\hat{x}_{LP}(y)$ and is it the unique solution? Arora et al. [ADS09] presented a certificate based on local structures both for $\hat{x}_{ML}(y)$ and $\hat{x}_{LP}(y)$ over a binary symmetric channel. In this section we present modifications of definitions and certificates to the case of memoryless binary-input output-symmetric (MBIOS) channels.

Notation: Let $y \in \mathbb{R}^n$ denote the received word. Let $\lambda = \lambda(y)$ denote the LLR vector for $y$. Let $x \in C(G)$ be a candidate for $\hat{x}_{ML}(y)$ and $\hat{x}_{LP}(y)$. $G$ is a $(d_{L}, d_{R})$-regular bipartite factor graph. For two vertices $u$ and $v$, denote by $d(u, v)$ the distance between $u$ and $v$ in $G$. Denote by $\mathcal{N}(v)$ the set of neighbors of a node $v$, and let $B(u, t)$ denote the set of vertices at distance at most $t$ from $u$.

Following Arora et al. we consider neighborhoods $B(i_0, 2T)$ where $i_0 \in V_L$ and $T < \frac{1}{4}girth(G)$. Note that the induced graph on $B(i_0, 2T)$ is a tree.

Definition 3 (Minimal Local Deviation, [ADS09]). An assignment $\beta \in \{0, 1\}^n$ is valid deviation of depth $T$ at $i_0 \in V_L$ or, in short, a $T$-local deviation at $i_0$, if $\beta_{i_0} = 1$ and $\beta$ satisfies all parity checks in $B(i_0, 2T)$,

$$\forall j \in V_R \cap B(i_0, 2T) : \sum_{i \in \mathcal{N}(j)} \beta_i \equiv 0 \mod 2.$$

A $T$-local deviation $\beta$ at $i_0$ is minimal if $\beta_i = 0$ for every $i \notin B(i_0, 2T)$, and every check node $j$ in $B(i_0, 2T)$ has at most two neighbors with value 1 in $\beta$. A minimal $T$-local deviation at $i_0$ can be seen as a subtree of $B(i_0, 2T)$ of height $2T$ rooted at $i_0$, where every variable node has full degree and every check node has degree 2. Such a tree is called a skinny tree. An assignment $\beta \in \{0, 1\}^n$ is a minimal $T$-local deviation if it is a minimal $T$-local deviation at some $i_0$. Note that given $\beta$ there is a unique such $i_0 \triangleq \text{root}(\beta)$.
If \( w = (w_1, \ldots, w_T) \in [0, 1]^T \) is a weight vector and \( \beta \) is a minimal \( T \)-local deviation, then \( \beta^{(w)} \) denotes the \( w \)-weighted deviation

\[
\beta^{(w)}_i = \begin{cases} 
  w_i \beta_i & \text{if } d(\text{root}(\beta), i) = 2t \text{ and } 1 \leq t \leq T, \\
  0 & \text{otherwise.}
\end{cases}
\]

The following definition expands the notion of addition of codewords over \( \mathbb{F}_2^n \) to the case where one of the vectors is fractional.

**Definition 4** ([Fel03]). Given a codeword \( x \in \{0, 1\}^n \) and a point \( f \in [0, 1]^n \), the relative point \( x \oplus f \in [0, 1]^n \) is defined by \( (x \oplus f)_i = |x_i - f_i| \).

Note that

\[
(x \oplus f)_i = \begin{cases} 
  1 - f_i & \text{if } x_i = 1, \\
  f_i & \text{if } x_i = 0.
\end{cases}
\]

Hence, for a fixed \( x \in \{0, 1\}^n \), \( x \oplus f \) is an affine linear function in \( f \). It follows that for any distribution over vectors \( f \in [0, 1]^n \), we have \( \mathbb{E}[x \oplus f] = x \oplus \mathbb{E}[f] \).

Given a log-likelihood ratio vector \( \lambda \), the cost of a \( w \)-weighted minimal \( T \)-local deviation \( \beta \) is defined by \( \langle \lambda, \beta^{(w)} \rangle \). The following definition is an extension of local optimality from BSC to LLR.

**Definition 5** (Local optimality following [ADS09]). A codeword \( x \in \{0, 1\}^n \) is \((T, w)\)-locally optimal for \( \lambda \in \mathbb{R}^n \) if for all minimal \( T \)-local deviations \( \beta \),

\[
\langle \lambda, x \oplus \beta^{(w)} \rangle > \langle \lambda, x \rangle.
\]

Since \( \beta^{(w)} \in [0, 1]^n \), we consider only weight vectors \( w \in [0, 1]^T \setminus \{0^n\} \). Koetter and Vontobel [KV06] proved for \( w = 1 \) that a locally optimal codeword \( x \) for \( \lambda \) is also globally optimal, i.e., the ML codeword. Moreover, they also showed that a locally optimal codeword \( x \) for \( \lambda \) is also the unique optimal LP solution given \( \lambda \). Arora et al. [ADS09] used a different technique to prove that local optimality is sufficient both for global optimality and LP optimality with
general weights in the case of a binary symmetric channel. We extend the results of Arora et al. [ADS09] to the case of MBIOS channels. Specifically, we prove for MBIOS channels that local optimality implies LP optimality (Theorem 9). We first show how to extend the proof that local optimality implies ML to MBIOS channels.

**Theorem 6** (local optimality is sufficient for ML). Let $T < \frac{1}{4} \text{girth}(G)$ and $w \in [0, 1]^T$. Let $\lambda \in \mathbb{R}^n$ denote the log-likelihood ratio for the received word, and suppose that $x \in \{0, 1 \}^n$ is a $(T, w)$-locally optimal codeword in $C(G)$ for $\lambda$. Then $x$ is also the unique maximum-likelihood codeword for $\lambda$.

The proof for MBIOS channels is a straightforward modification of the proof in [ADS09]. We include it for the sake of self-containment. The following Lemma is the key structural lemma in the proof of Theorem 6.

**Lemma 7** ([ADS09]). Let $T < \frac{1}{4} \text{girth}(G)$. Then, for every codeword $z \neq 0^n$, there exists a distribution over minimal $T$-local deviations $\beta$ such that, for every weight vector $w \in [0, 1]^T$, there exists an $\alpha \in (0, 1]$, such that

$$
E_\beta \beta^w = \alpha z.
$$

**Proof of Theorem 6**. We want to show that for every codeword $x' \neq x$, $\langle \lambda, x' \rangle > \langle \lambda, x \rangle$. Since $z \triangleq x \oplus x'$ is a codeword, by Lemma 7 there exists a distribution over minimal $T$-local deviations $\beta$ such that $E_\beta \beta^w = \alpha z$. Let $f : [0, 1]^n \rightarrow \mathbb{R}$ be the affine linear function defined by $f(u) \triangleq \langle \lambda, x \oplus u \rangle = \langle \lambda, x \rangle + \sum_{i=1}^n (-1)^{x_i} \lambda_i u_i$. Then,

\[
\langle \lambda, x \rangle < E_\beta \langle \lambda, x \oplus \beta^w \rangle \quad \text{(by local optimality of $x$)}
\]

\[
= \langle \lambda, x \oplus E_\beta \beta^w \rangle \quad \text{(by linearity of $f$ and linearity of expectation)}
\]

\[
= \langle \lambda, x \oplus \alpha z \rangle \quad \text{(by Lemma 7)}
\]

\[
= \langle \lambda, (1 - \alpha)x + \alpha(x \oplus z) \rangle
\]

\[
= \langle \lambda, (1 - \alpha)x + \alpha x' \rangle
\]

\[
= (1 - \alpha) \langle \lambda, x \rangle + \alpha \langle \lambda, x' \rangle.
\]
which implies that $\langle \lambda, x' \rangle > \langle \lambda, x \rangle$ as desired.

In order to prove a sufficient condition for LP optimality, we consider graph cover decoding introduced by Vontobel and Koetter \[VK05\]. We use the terms and notation of Vontobel and Koetter \[VK05\] in the statement of Lemma 8 and the proof of Theorem 9 (See Appendix A). The following lemma shows that local optimality is preserved after lifting to an $M$-cover. Note that the weight vector must be scaled by the cover degree $M$.

**Lemma 8.** Let $T < \frac{1}{4}girth(G)$ and $w \in [0, \frac{1}{M}]^T \setminus \{0^n\}$. Let $\tilde{G}$ denote any $M$-cover of $G$. Suppose that $x \in C(G)$ is a $(T, w)$-locally optimal codeword for $\lambda \in \mathbb{R}^n$. Let $\tilde{x} = x^M \in C(\tilde{G})$ and $\tilde{\lambda} = \lambda^M \in \mathbb{R}^{n \cdot M}$ denote the $M$-lifts of $x$ and $\lambda$, respectively. Then $\tilde{x}$ is a $(T, M \cdot w)$-locally optimal codeword for $\tilde{\lambda}$.

**Proof.** Assume that $\tilde{x} = x^M$ is not a $(T, M \cdot w)$-locally optimal codeword for $\tilde{\lambda} = \lambda^M$. Then, there exists a minimal $T$-local deviation $\tilde{\beta} \in \{0, 1\}^{n \cdot M}$ such that

$$\langle \tilde{\lambda}, \tilde{x} \oplus \tilde{\beta}^{(M \cdot w)} \rangle \leq \langle \tilde{\lambda}, \tilde{x} \rangle.$$  

(3)

Note that for every $\tilde{x} \in \{0, 1\}^{n \cdot M}$ and its projection $x \in \mathbb{R}^n$, it holds that

$$\frac{1}{M} \langle \tilde{\lambda}, \tilde{x} \rangle = \langle \lambda, x \rangle,$$

and

$$\frac{1}{M} \langle \tilde{\lambda}, \tilde{x} \oplus \tilde{\beta}^{(M \cdot w)} \rangle = \langle \lambda, x \oplus \beta^{(w)} \rangle,$$

(4)

(5)

where $\beta$ is the support of the projection of $\tilde{\beta}$ onto the base graph. $\beta$ is a local deviation because $T \leq \frac{1}{4}girth(G) \leq \frac{1}{4}girth(\tilde{G})$. From Equations (3), (4), and (5) we get that $\langle \lambda, x \rangle \geq \langle \lambda, x \oplus \beta^{(w)} \rangle$, contradicting our assumption on the $(T, w)$-local optimality of $x$. Therefore, $\tilde{x}$ is a $(T, M \cdot w)$-locally optimal codeword for $\tilde{\lambda}$. \hfill \Box

Arora et al. \[ADS09\] proved the following theorem for a BSC and $w \in [0, 1]^T$. The proof can be extended to the case of MBIOS channels with $w \in [0, 1]^T$ using the same technique of Arora et al. A simpler proof is achieved for $w \in [0, \frac{1}{M}]^T$ for some finite $M$. The proof is based
on arguments utilizing properties of graph cover decoding [VK05], and follows as a corollary of Theorem 6 and Lemma 8.

**Theorem 9** (local optimality is sufficient for LP optimality). For every factor graph $G$, there exists a constant $M$ such that, if

1. $T < \frac{1}{4} \text{girth}(G)$,
2. $w \in [0, \frac{1}{M}]^T \setminus \{0\}^T$, and
3. $x$ is a $(T, w)$-locally optimal codeword for $\lambda \in \mathbb{R}^n$,

then $x$ is also the unique optimal LP solution given $\lambda$.

**Proof.** Suppose that $x$ is a $(T, w)$-locally optimal codeword for $\lambda \in \mathbb{R}^n$. Vontobel and Koetter [VK05] proved that for every basic feasible solution $z \in [0, 1]^n$ of the LP, there exists an $M$-cover $\tilde{G}$ of $G$ and an assignment $\tilde{z} \in \{0, 1\}^{n-M}$ such that $\tilde{z} \in C(\tilde{G})$ and $z = p(\tilde{z})$, where $p(\tilde{z})$ is the image of the scaled projection of $\tilde{z}$ in $G$ (i.e., pseudo-codeword associated with $\tilde{z}$). Moreover, since the number of basic feasible solutions is finite, we conclude that there exists a finite $M$-cover $\tilde{G}$ such that every basic feasible solution of the LP admits an eligible assignment in $\tilde{G}$.

Let $z^*$ denote an optimal LP solution given $\lambda$. Without loss of generality $z^*$ is a basic feasible solution. Let $\tilde{z}^* \in \{0, 1\}^{n-M}$ denote the $0-1$ assignment in the $M$-cover $\tilde{G}$ that corresponds to $z^* \in [0, 1]^n$. By the equivalence of LP decoding and graph cover decoding [VK05], Equation (4) and the optimality of $z^*$ it follows that $\tilde{z}^*$ is a codeword in $C(\tilde{G})$ that minimizes $\langle \tilde{\lambda}, \tilde{z} \rangle$ for $\tilde{z} \in C(\tilde{G})$, namely $\tilde{z}^* = \hat{x}^{ML}(y^{\uparrow M})$.

Let $\tilde{x} = x^{\uparrow M}$ denote the $M$-lift of $x$. Lemma 8 implies that $\tilde{x}$ is a $(T, M \cdot w)$-locally optimal codeword for $\tilde{\lambda}$, where $M \cdot w \in [0, 1]^T$. By Theorem 6 we also get that $\tilde{x} = \hat{x}^{ML}(y^{\uparrow M})$. Moreover, Theorem 6 guarantees the uniqueness of an ML optimal solution. Thus, $\tilde{x} = \tilde{z}^*$. By projection to $G$, since $\tilde{x} = \tilde{z}^*$, we get that $x = z^*$ and uniqueness follows, as required. \qed

\[From this point, let $M$ denote the constant, the existence of which is guaranteed by Theorem 9.\]
4 Proving Error Bounds Using Local Optimality

In order to simplify the probabilistic analysis of algorithms for decoding linear codes over symmetric channels, it is common to assume that the all-zeros codeword is the transmitted codeword, i.e., \( x = 0^n \). The following lemma gives a structural characterization for the event of LP decoding failure if \( x = 0^n \).

**Lemma 10.** Let \( T < \frac{1}{4} \text{girth}(G) \). Assume that the all-zeros codeword was transmitted, and let \( \lambda \in \mathbb{R}^n \) denote the log-likelihood ratio for the received word. If the LP decoder fails to decode the all-zeros codeword, then for every \( w \in \mathbb{R}^T_+ \) there exists a minimal \( T \)-local deviation \( \beta \) such that \( \langle \lambda, \beta(w) \rangle \leq 0 \).

**Proof.** Consider the event where the LP decoder fails to decode the all-zeros codeword, i.e., \( 0^n \) is not a unique optimal LP solution. Theorem 9 implies that there exists a constant \( M \) such that, for every \( w' \in [0, \frac{1}{M}]^T \setminus \{0^T\} \), the all-zeros codeword is not \((T, w')\)-locally optimal codeword for \( \lambda \). That is, there exists a minimal \( T \)-local deviation \( \beta \) such that \( \langle \lambda, \beta(w') \rangle \leq 0 \). Let \( w' = \frac{1}{M \cdot \|w\|_\infty} \cdot w \). Therefore \( \langle \lambda, \beta(w) \rangle \) is also non-positive, as required. \( \square \)

Note that the correctness of the all-zeros assumption depends on the employed decoding algorithm. Although this assumption is trivial for ML decoding because of the symmetry of a linear code \( C(G) \), it is not immediately clear in the context of LP decoding. Feldman et al. [Fel03, FWK05] noticed that the fundamental polytope \( P(G) \) is highly symmetric, and proved the following theorem.

**Theorem 11** (All-zeros assumption, [FWK05]). The probability that the LP decoder fails is independent of the transmitted codeword.

Therefore, one can assume that \( x = 0^n \) when analyzing LP-decoding failure for linear codes. The following corollary follows Lemma 10 and Theorem 11.

**Corollary 12.** Fix \( T < \frac{1}{4} \text{girth}(G) \) and \( w \in \mathbb{R}^T_+ \). Then,

\[
P\{\text{LP decoding fails} \} \leq P\{ \exists \beta. \langle \lambda, \beta(w) \rangle \leq 0 | x = 0^n \}.
\]
4.1 Bounding Processes on Trees

Using the terminology of Corollary \cite{ADS09}, Arora et al. suggested a recursive method for bounding the probability \( P\{\exists \beta. (\lambda, \beta^{(w)}) \leq 0 | x = 0^n\} \) for a BSC. We extend this method to MBIOS channels and apply it to a BI-AWGN channel.

Let \( G \) be a \((d_L, d_R)\)-regular bipartite factor graph, and fix \( T < \frac{1}{4} \text{girth}(G) \). Let \( T_{v_0} \) denote the subgraph induced by \( B(v_0, 2T) \) for a variable node \( v_0 \). Since \( T < \frac{1}{4} \text{girth}(G) \), it follows that \( T_{v_0} \) is a tree. We direct the edges of \( T_{v_0} \) so that it is an out-branching directed at the root \( v_0 \).

For \( l \in \{0, \ldots, 2T\} \), denote by \( V_l \) the set of vertices of \( T_{v_0} \) at height \( l \) (the leaves have height 0 and the root has height \( 2T \)). Let \( \tau \subseteq V(T_{v_0}) \) denote the vertex set of a skinny tree rooted at \( v_0 \).

**Definition 13** \(((T, \omega))\)-Process on a \((d_L, d_R)\)-Tree, \cite{ADS09}. Let \( \omega \in \mathbb{R}_+^T \) denote a weight vector. Let \( \lambda \) denote an assignment of real values to the variable nodes of \( T_{v_0} \), we define the \( \omega \)-weighted value of a skinny tree \( \tau \) by

\[
\text{val}_\omega(\tau; \lambda) \triangleq \sum_{l=0}^{T-1} \sum_{v \in \tau \cap V_{2^l}} \omega_l \cdot \lambda_v.
\]

Namely, the sum of the values of variable nodes in \( \tau \) weighted according to their height.

Given a probability distribution over assignments \( \lambda \), we are interested in the probability

\[
\Pi_{\lambda, d_L, d_R}(T, \omega) \triangleq \mathbb{P}_\lambda \left\{ \min_{\tau \subset T} \text{val}_\omega(\tau; \lambda) \leq 0 \right\}.
\]

In other words, \( \Pi_{\lambda, d_L, d_R}(T, \omega) \) is the probability that the minimum value over all skinny trees of height \( 2T \) rooted in some variable node \( v_0 \) in a \((d_L, d_R)\)-bipartite graph \( G \) is non-positive. For every two roots \( v_0 \) and \( v_1 \) the trees \( T_{v_0} \) and \( T_{v_1} \) are isomorphic, it follows that \( \Pi_{\lambda, d_L, d_R}(T, \omega) \) does not depend on the root \( v_0 \).

Since \( \lambda \) is a random assignment of values to variable nodes in \( T_{v_0} \), Arora et al. refer to \( \min_{\tau \subset T_{v_0}} \text{val}_\omega(\tau; \lambda) \) as a random process. With this notation, we apply a union bound utilizing Lemma \[10\] as follows.

**Lemma 14.** Let \( G \) be a \((d_L, d_R)\)-regular bipartite graph and \( w \in \mathbb{R}_+^T \) be a weight vector
with $T < \frac{1}{2} \text{girth}(G)$. Suppose that $\lambda \in \mathbb{R}^n$ is the log-likelihood ratio of the word received from the channel. Then, the transmitted codeword $x = 0^n$ is $(T, \alpha \cdot w)$-locally optimal for

$$\alpha \triangleq (M \cdot \|w\|_{\infty})^{-1}$$

with probability at least

$$1 - n \cdot \Pi_{\lambda,d_L,d_R}(T, \omega), \quad \text{where } \omega_l = w_{T-l},$$

and with at least the same probability, $x = 0^n$ is also the unique optimal LP solution given $\lambda$.

Note the two different weight notations: (i) $w$ denotes weight vector in the context of weighted deviations, and (ii) $\omega$ denotes weight vector in the context of skinny subtrees in the $(T, \omega)$-Process. A one-to-one correspondence between these two vectors is given by $\omega_l = w_{T-l}$ for $0 \leq l < T$. From this point on, we refer only to the notation of $\omega$.

Following Lemma [4], it is sufficient to estimate the probability $\Pi_{\lambda,d_L,d_R}(T, \omega)$ for a given weight vector $\omega$, a distribution of a random vector $\lambda$, and degrees $(d_L, d_R)$. We overview the recursion presented in [ADS09] for estimating and bounding the probability of the existence of a skinny tree with non-positive value in a $(T, \omega)$-process.

Let $\{\gamma\}$ denote an ensemble of i.i.d. random variables. Define random variables $X_0, ..., X_{T-1}$ and $Y_0, ..., Y_{T-1}$ with the following recursion:

$$Y_0 = \omega_0 \gamma$$

$$X_l = \min \{Y_l^{(1)}, ..., Y_l^{(d_R-1)}\} \quad (0 \leq l < T)$$

$$Y_l = \omega_l \gamma + X_{l-1}^{(1)} + ... + X_{l-1}^{(d_L-1)} \quad (0 < l < T)$$

The notation $X_1^{(1)}, ..., X_d^{(d)}$ denotes $d$ mutually independent copies of the random variable $X$. Each instance of $Y_l$, $0 \leq l < T$, uses an independent instance of a random variable $\gamma$.

Consider a directed tree $\mathcal{T} = \mathcal{T}_{v_0}$ of height $2T$, rooted at node $v_0$. Associate variable nodes of $\mathcal{T}$ at height $2l$ with copies of $Y_l$, and check nodes at height $2l + 1$ with copies of $X_l$, for $0 \leq l < T$. Note that any realization of the random variables $\{\gamma\}$ to variable nodes in $\mathcal{T}$ can be viewed as an assignment $\lambda$. Thus, the minimum value of a skinny tree of $\mathcal{T}$
equals $\sum_{i=1}^{d_L} X_{T-1}^{(i)}$. This implies that the recursion in Equations (7)-(9) defines a dynamic programming algorithm for computing $\min_{\tau \subseteq T} \text{val}_{\omega}(\tau; \lambda)$. Now, let the components of the LLR vector $\lambda$ be i.i.d. random variables distributed identically to $\{\gamma\}$, then

$$\Pi_{\lambda,d_L,d_R}(T, \omega) = P\left\{ \sum_{i=1}^{d_L} X_{T-1}^{(i)} \leq 0 \right\}. \quad (10)$$

Given a distribution of $\{\gamma\}$ and a finite “height” $T$, it is possible to compute the distribution of $X_L$ and $Y_L$ according to the recursion in Equations (7)-(9) using properties of a sum of random variables and a minimum of random variables (see Appendix B.1). The following two lemmas play a major role in proving bounds on $\Pi_{\lambda,d_L,d_R}(T, \omega)$.

**Lemma 15 ([ADS09]).** For every $t \geq 0$,

$$\Pi_{\lambda,d_L,d_R}(T, \omega) \leq \left( E e^{-tX_{T-1}} \right)^{d_L}. \quad (11)$$

Let $d'_L \triangleq d_L - 1$ and $d'_R \triangleq d_R - 1$.

**Lemma 16 ([ADS09]).** For $0 \leq s < l < T$, we have

$$E e^{-tX_l} \leq \left( E e^{-tX_s} \right)^{d'_L l-s} \cdot \prod_{k=0}^{l-s-1} \left( d'_R E e^{-t\omega_{l-k}} \right)^{d'_L k}. \quad (12)$$

Based on these bounds, in the following subsection we present concrete bounds on $\Pi_{\lambda,d_L,d_R}(T, \omega)$ for BI-AWGN channel.

### 4.2 Analysis for BI-AWGN Channel

Consider the binary input additive white Gaussian noise channel with noise variance $\sigma^2$ denoted by BI-AWGNC($\sigma$). In the case that the codeword is all zeros, the channel input is $X_i = +1$ for every $i$. Hence, $\lambda_{i}^{BI-AWGNC(\sigma)} = \frac{2}{\sigma^2}(1 + \phi_i)$ where $\phi_i \sim N(0, \sigma^2)$. Since $\Pi_{\lambda,d_L,d_R}(T, \omega)$ is invariant under positive scaling of the vector $\lambda$, we consider in the following analysis the scaled vector $\lambda$ in which $\lambda_i = 1 + \phi_i$ with $\phi_i \sim N(0, \sigma^2)$.  

16
Following [ADS09], we apply a simple analysis for BI-AWGNC(σ) with uniform weight vector ω. Then, we present improved bounds by using a non-uniform weight vector.

4.2.1 Uniform Weights

Consider the case where ω = {1}^T. Let c_1 ≜ Ee^{-tX_0} and c_2 ≜ d'_R E e^{-tλ_i}, and define c ≜ c_1 \cdot c_2^{1/(d_L-1)}. By substituting notations of c_1 and c_2 in Lemmas 15 and 16, Arora et al. [ADS09] proved that if c < 1, then

\[ \Pi_{\lambda,d_L,d_R}(T, \{1\}^T) \leq c^{d_L - d'_L} e^{-d_L - d_L}. \]

To analyze parameters for which \( \Pi_{\lambda,d_L,d_R}(T, \{1\}^T) \to 0 \), we need to compute \( c_1 \) and \( c_2 \) as functions of σ, \( d_L \) and \( d_R \). Note that

\[ X_0 = \min_{i \in [d'_R]} \{ \lambda_i \} = 1 + \min_{i \in [d'_R]} \phi_i, \text{ where } \phi_i \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d.} \]

Denote by \( f_N(\cdot) \) and \( F_N(\cdot) \) the p.d.f. and c.d.f. of a Gaussian random variable with zero mean and standard deviation σ, respectively. We therefore have

\begin{align*}
  c_1(\sigma, d_L, d_R) &= d'_R e^{-t} \int_{-\infty}^{\infty} \left(1 - F_N(x)\right)^{d'_R - 1} f_N(x)e^{-tx}dx, \quad \text{and} \quad \quad (11) \\
  c_2(\sigma, d_L, d_R) &= d'_R e^{\frac{1}{2}t^2\sigma^2 - t}. \quad \quad (12)
\end{align*}

The above calculations give the following bound on \( \Pi_{\lambda,d_L,d_R}(T, \{1\}^T) \).

**Lemma 17.** If σ > 0 and \( d_L, d_R > 2 \) satisfy the condition

\[ c = \min_{t \geq 0} \left( \frac{d'_R e^{-t} \int_{-\infty}^{\infty} \left(1 - F_N(x)\right)^{d'_R - 1} f_N(x)e^{-tx}dx}{c_1} \right) \cdot \left( \frac{d'_R e^{\frac{1}{2}t^2\sigma^2 - t}}{c_2} \right)^{1/(d_L-2)} < 1, \]
then for $T \in \mathbb{N}$ and $\omega = 1^T$, we have

$$
\Pi_{\lambda,d_L,d_R}(T, \omega) \leq c^{d_L \cdot d_L^{T-1} - d_L}.
$$

For (3,6)-regular graphs, we obtain by numeric calculations the following corollary.

**Corollary 18.** Let $\sigma < 0.59$, $d_L = 3$, and $d_R = 6$. Then, there exists a constant $c < 1$ such that for every $T \in \mathbb{N}$ and $\omega = \{1\}^T$,

$$
\Pi_{\lambda,d_L,d_R}(T, \omega) \leq c^{2^T}.
$$

### 4.2.2 Improved Bounds Using Non-Uniform Weights

The following lemma implies an improved bound for $\Pi_{\lambda,d_L,d_R}(T, \omega)$ using non-uniform weight vector $\omega$.

**Lemma 19.** Let $\sigma > 0$ and $d_L, d_R > 2$. Suppose that for some $s \in \mathbb{N}$ and some weight vector $\overline{\omega} \in \mathbb{R}^s_+$,

$$
\min_{t \geq 0} \mathbb{E} e^{-tX_s} < \left((d_R - 1) e^{-\frac{1}{2\sigma^2}}\right)^{-\frac{1}{d_L} - \frac{1}{d_L}}.
$$

Let $\omega^{(\rho)} \in \mathbb{R}^T_+$ denote the concatenation of the vector $\overline{\omega} \in \mathbb{R}^s_+$ and the vector $(\rho, ..., \rho) \in \mathbb{R}^{T-s}$. Then, for every $T > s$ there exist constants $c < 1$ and $\rho \geq 0$ such that

$$
\Pi_{\lambda,d_L,d_R}(T, \omega^{(\rho)}) \leq \left((d_R - 1) e^{-\frac{1}{2\sigma^2}}\right)^{-\frac{d_L}{d_L^s - 2}} \cdot c^{d_L \cdot d_L^{T-s-1}}.
$$

Note that $\Pi_{\lambda,d_L,d_R}(T, \omega^{(\rho)})$ decreases doubly exponential as a function of $T$.

**Proof.** By Lemma 16 we have

$$
\mathbb{E} e^{-tX_{T-1}} \leq (\mathbb{E} e^{-tX_s})^{(d_L - 1)^{T-s-1}} \left((d_R - 1) \mathbb{E} e^{-t_{\rho}(1+\phi)}\right)^{\sum_{k=0}^{T-s-2} (d_L - 1)^k}
$$

$$
= (\mathbb{E} e^{-tX_s})^{(d_L - 1)^{T-s-1}} \left((d_R - 1) \mathbb{E} e^{-t_{\rho}(1+\phi)}\right)^{(d_L - 1)^T - 1} d_L^{T-s-1}.
$$

Note that $\mathbb{E} e^{-t_{\rho}(1+\phi)} = e^{-t_{\rho} + \frac{1}{2\sigma^2} t_{\rho}^2}$ is minimized when $t_{\rho} = \sigma^{-2}$. By setting $\rho = \frac{1}{t_{\sigma^2}}$, we
obtain
\[
\mathbb{E} e^{-tX_{T-1}} \leq (\mathbb{E} e^{-tX_s})^{(d_L-1)^T-s-1} \left( (d_R - 1) e^{-\frac{1}{2\sigma^2}} \right)^{(d_L-1)^T-s-1} \frac{(d_L-1)^T-s-1}{d_L-2}.
\]

Let \( c \triangleq \min_{t \geq 0} \mathbb{E} e^{-tX_s} \). By Equation (13), \( c < 1 \). Let \( t^* = \arg \min_{t \geq 0} \mathbb{E} e^{-tX_s} \), then
\[
\mathbb{E} e^{-t^*X_{T-1}} \leq c^{(d_L-1)^T-s-1} \left( (d_R - 1) e^{-\frac{1}{2\sigma^2}} \right)^{\frac{1}{d_L-2}}.
\]

Using Lemma 15, we conclude
\[
\Pi_{\lambda,d_L,d_R}(T, \omega(\rho)) \leq c^{d_L(d_L-1)^T-s-1} \left( (d_R - 1) e^{-\frac{1}{2\sigma^2}} \right)^{\frac{1}{d_L-2}},
\]
and the lemma follows. 

Arora et al. [ADS09] suggested using a weight vector \( \overline{\omega} \) with components \( \overline{\omega}_l = (d_L - 1)^l \). This weight vector has the effect that if \( \lambda \) assigns the same value to every variable node, then every level in a skinny tree \( \tau \) contributes equally to \( \text{val}_{\overline{\omega}}(\tau; \lambda) \). For \( T > s \), consider a weight vector \( \omega(\rho) \in \mathbb{R}_{+}^T \) defined by
\[
\omega_l = \begin{cases} 
\overline{\omega} & \text{if } 0 \leq l < s, \\
\rho & \text{if } s \leq l < T.
\end{cases}
\]
Note that the first \( s \) components of \( \omega(\rho) \) are non-uniform while the other components are uniform.

For a given \( \sigma, d_L, \) and \( d_R, \) and for a concrete value \( s \) we can compute the distribution of \( X_s \) using the recursion in Equations (7)-(9). Moreover, we can also compute the value \( \min_{t \geq 0} \mathbb{E} e^{-tX_s} \). Computing the distribution and the Laplace transform of \( X_s \) is not a trivial task in the case where the components of \( \lambda \) have a continuous density distribution function. However, since the Gaussian distribution function is smooth and most of its volume is concentrated in a defined interval, it is possible to "simulate" the evolution of the density distribution.
functions of the random variables $X_i$ and $Y_i$ for $i \leq s$. We use numeric method based on quantization in order to represent and evaluate the functions $f_{X_i}(\cdot)$, $F_{X_i}(\cdot)$, $f_{Y_i}(\cdot)$, and $F_{Y_i}(\cdot)$. This computation follows methods used in the implementation of density evolution technique (see e.g. [RU08]). A specific method for computation is described in Appendix B and exemplified for (3,6)-regular graphs.

For (3, 6)-regular bipartite graphs we obtain the following corollary.

**Corollary 20.** Let $\sigma < \sigma_0$, $d_L = 3$, and $d_R = 6$. For the following values of $\sigma_0$ and $s$ in Table 1 it holds that there exists a constant $c < 1$ such that for every $T > s$,

$$
\Pi_{\lambda,d_L,d_R}(T, \omega) \leq \frac{1}{125}e^{-\frac{3}{2}\sigma^2} \cdot c^{2T-s-1}.
$$

Theorem 1 follows Lemma 14, Lemma 17, and Corollary 20 by taking $T = \Theta(\log n) < \frac{1}{4}girth(G)$. Theorem 2 is obtained in the same manner after a simple straightforward modification of Lemma 17 to MBIOS channels.

**Remark:** Following [ADS09], the contribution $\omega_T \cdot \lambda_{v_0}$ of the root of $T_{v_0}$ is not included in the definition of $val_\omega(\tau; \lambda)$. The effect of this contribution to $\Pi_{\lambda,d_L,d_R}(T, \omega)$ is bounded by a multiplicative factor, as implied by the proof of Lemma 15. The multiplicative factor is bounded by $E e^{-\omega_T \lambda_{v_0}}$, which may be regarded as a constant since it does not depend on the code parameters (in particular the code length $n$).

<table>
<thead>
<tr>
<th>$s$</th>
<th>$\sigma_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.605</td>
</tr>
<tr>
<td>1</td>
<td>0.635</td>
</tr>
<tr>
<td>2</td>
<td>0.66</td>
</tr>
<tr>
<td>3</td>
<td>0.675</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$s$</th>
<th>$\sigma_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.685</td>
</tr>
<tr>
<td>6</td>
<td>0.7</td>
</tr>
<tr>
<td>8</td>
<td>0.71</td>
</tr>
<tr>
<td>10</td>
<td>0.715</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$s$</th>
<th>$\sigma_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.72</td>
</tr>
<tr>
<td>14</td>
<td>0.725</td>
</tr>
<tr>
<td>18</td>
<td>0.73</td>
</tr>
<tr>
<td>22</td>
<td>0.735</td>
</tr>
</tbody>
</table>

Table 1: Threshold values $\sigma_0$ for finite $s$ in Corollary 20.
5 Discussion

We extended the analysis of Arora et al. [ADS09] for LP decoding over BSC to any MBIOS channel. We proved a condition that guarantees an exponentially small probability for decoding error with respect to LP decoding of regular LDPC codes with logarithmic girth over MBIOS channels. We also proved thresholds for exponentially small error probability with respect to LP decoding of regular LDPC codes with logarithmic girth over binary-input AWGN channels.

Although thresholds are usually regarded as an asymptotic result, the analysis presented by Arora et al. [ADS09], as well as its extension presented in this paper, exhibits both asymptotic results as well as finite results. An interesting tradeoff between these two perspectives is shown by the formulation of the results. We regard the goal of achieving the highest possible thresholds as an asymptotic goal, and as such we may compare the achieved thresholds to the asymptotic BP thresholds. Note that the obtained threshold increases up to a certain ceiling value (LP threshold) as the girth increases. Thus, an asymptotic result is obtained.

However, in the case of finite length codes a finite analysis cannot assume that the girth tends to infinity. Two phenomena occur in the analysis of finite codes: (i) the threshold increases as function of the girth (as shown in Table 1), and (ii) the decoding error probability decreases exponentially as a function of the gap $\text{threshold} - \sigma$ (as implied by Figure 5(b)). We demonstrate the power of the analysis for the finite case by presenting thresholds of $\sigma = 0.605$ provided that the girth is at least 4, and $\sigma = 0.735$ for girth greater than 88 for (3, 6)-regular LDPC codes.

In the proof of LP optimality (Lemma 8 and Theorem 9) we used the combinatorial interpretation of LP decoding via graph covers [VK05] to infer a reduction to conditions of ML optimality. That is, the decomposition of codewords presented by Arora et al. [ADS09] leads to a decomposition for fractional LP solutions. This method of reducing combinatorial characterizations of LP decoding to combinatorial characterizations of ML decoding is based on graph cover decoding.
Future directions: The technique for proving error bounds for BI-AWGN channel described in Section 4 and in Appendix B is based on a min-sum probabilistic process on a tree. The process is characterized by an evolution of probability density functions. Computing the evolving densities in the analysis of AWGN channels is not a trivial task. As indicated by our numeric computations, the evolving density functions in the case of the AWGN channel visually resemble Gaussian probability density functions (see Figures 2 and 3). Chung et al. [CRU01] presented a method for estimating thresholds of belief propagation decoding according to density evolution using Gaussian approximation. Applying an appropriate Gaussian approximation technique to our analysis may result with analytic asymptotic approximate thresholds of LP decoding for regular LDPC codes over AWGN channels.

Feldman et al. [FKV05] observed that for high SNRs truncating LLRs of BI-AWGN surprisingly assist LP decoding. They proved that for certain families of regular LDPC codes and large enough SNRs (i.e., small $\sigma$), it is advantageous to truncate the LLRs before passing them to the LP decoder. The method presented in Appendix B for computing densities evolving on trees using quantization and truncation of the LLRs can be applied to this case. It is interesting to see whether this unexpected phenomenon of LP decoding occurs also for larger values of $\sigma$ (i.e., lower SNRs).

A Graph Cover Decoding - Basic Terms and Notation

Vontobel and Koetter introduced in [VK05] a combinatorial concept called graph-cover decoding (GCD) for decoding codes on graphs, and showed its equivalence to LP decoding. The characterization of GCD provides a useful theoretical tool for the analysis of LP decoding and its connections to iterative message-passing decoding algorithms. We use the characterization of graph cover decoding in the statement of Lemma 8 and the proof of Theorem 9. In the following, we define few basic terms and notation of graph covers and graph cover decoding.

Let $G$ and $\tilde{G}$ be finite graphs and let $\pi : \tilde{G} \to G$ be a graph homomorphism, namely, $\forall \tilde{u}, \tilde{v} \in V(\tilde{G}) : (\tilde{u}, \tilde{v}) \in E(\tilde{G}) \Rightarrow (\pi(\tilde{u}), \pi(\tilde{v})) \in E(G)$. A homomorphism $\pi$ is a covering
map if for every $\tilde{v} \in V(\tilde{G})$ the restriction of $\pi$ to neighbors of $\tilde{v}$ is a bijection to the neighbors of $\pi(\tilde{v})$. The pre-image $\pi^{-1}(v)$ of a node $v$ is called a fiber and is denoted by $\tilde{G}_v$. It is easy to see that all the fibers have the same cardinality if $G$ is connected. This common cardinality is called the degree or fold number of the covering map. If $\pi : \tilde{G} \rightarrow G$ is a covering map, we call $G$ the base graph and $\tilde{G}$ a cover of $G$. In the case where the fold number of the covering map is $M$, we say that $\tilde{G}$ is an $M$-cover of $G$.

Given a base graph $G$ and a natural fold number $M$, an $M$-cover $\tilde{G}$ and a covering map $\pi : \tilde{G} \rightarrow G$ can be constructed in the following way. Map every vertex $(v, i) \in V(\tilde{G})$ (where $i \in [M]$) to $v \in V(G)$, i.e., $\pi(v, i) = v$. The edges in $E(\tilde{G})$ are obtained by specifying a matching $D_{(u, v)}$ of $M$ edges between $\pi^{-1}(u)$ and $\pi^{-1}(v)$ for every $(u, v) \in E(G)$.

Note that the term ‘covering’ originates from covering maps in topology, as opposed to other notions of ‘coverings’ in graphs or codes (e.g., vertex covers or covering codes).

We now define assignments to variable nodes in an $M$-cover of a Tanner graph. The assignment is induced by the covering map and an assignment to the variable nodes in the base graph.

**Definition 21** (lift, [VK05]). Consider a bipartite graph $G = (\mathcal{I} \cup \mathcal{J}, E)$ and an arbitrary $M$-cover $\tilde{G} = (\tilde{\mathcal{I}} \cup \tilde{\mathcal{J}}, \tilde{E})$ of $G$. The $M$-lift of a vector $x \in \mathbb{R}^N$ is an assignment $\tilde{x} \in \mathbb{R}^{N \cdot M}$ to the nodes in $\tilde{\mathcal{I}}$ that is induced by the assignment $x \in \mathbb{R}^N$ to the nodes in $\mathcal{I}$ and the covering map $\pi : \tilde{G} \rightarrow G$ as follows: Every $\tilde{v} \in \pi^{-1}(v)$ is assigned by $\tilde{x}$ the value assigned to $v$ by $x$. The $M$-lift of a vector $x$ is denoted by $x^{1M}$.

**Definition 22** (pseudocodeword, [VK05]). The (scaled) pseudocodeword $p(\tilde{x}) \in \mathbb{Q}^N$ associated with binary vector $\tilde{x} = \{\tilde{x}_{\tilde{v}}\}_{\tilde{v} \in \tilde{\mathcal{I}}} \in \tilde{C}$ of length $N \cdot M$ is the rational vector $p(\tilde{x}) \triangleq (p_1(\tilde{x}), p_2(\tilde{x}), ..., p_N(\tilde{x}))$ defined by

$$p_i(\tilde{x}) \triangleq \frac{1}{M} \cdot \sum_{\tilde{v} \in \pi^{-1}(v_i)} \tilde{x}_{\tilde{v}}, \quad (14)$$

where the sum is taken in $\mathbb{R}$ (not in $\mathbb{F}_2$).
B Computing the Evolution of Probability Densities over Trees

In this section we present a computational method for estimating \( \min_{t \geq 0} E^e^{-tX_s} \) for some concrete \( s \). The random variable \( X_s \) is defined by the recursion in Equations (7)-(9). Let \( \{ \gamma \} \) denote an ensemble of i.i.d. continuous random variable with probability density function (p.d.f.) \( f_\gamma(\cdot) \) and cumulative distribution function (c.d.f.) \( F_\gamma(\cdot) \).

We demonstrate the method for computing \( \min_{t \geq 0} E^e^{-tX_s} \) for the case where \( d_L = 3 \), \( d_R = 6 \), \( \omega_l = (d_L - 1)^l = 2^l \), \( \sigma = 0.7 \), and \( \gamma = 1 + \phi \) where \( \phi \sim \mathcal{N}(0, \sigma^2) \). In this case,

\[
\begin{align*}
    f_\gamma(x) &= f_{\mathcal{N}}(x-1) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}}, \quad \text{and} \\
    F_\gamma(x) &= F_{\mathcal{N}}(x-1) = \frac{1}{2} \left[ 1 + \text{erf}\left( \frac{x-1}{\sqrt{2}\sigma} \right) \right],
\end{align*}
\]

where \( \text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \) denotes the error function.

The actual computation of the of the evolution of density functions via the recursion equations requires a numeric implementation. Finding an efficient and stable such implementation is nontrivial. We follow methods used in the computation of the variable-node update process in the implementation of density evolution analysis (see e.g. RU08).

We first state two properties of random variables for the evolving process defined in the recursion. We then show a method for computing a proper representation of the probability density function of \( X_s \) for the purpose of finding \( \min_{t \geq 0} E^e^{-tX_s} \).

B.1 Properties of Random Variables

**Sum of Random Variables.** Let \( \Phi \) denote a random variable that equals to the sum of \( n \) independent random variables \( \{ \phi_i \}_{i=1}^n \), i.e., \( \Phi = \sum_{i \in [n]} \phi_i \). Denote by \( f_{\phi_i}(\cdot) \) the p.d.f. of \( \phi_i \). Then, the p.d.f. of \( \Phi \) is given by

\[
    f_\Phi = \bigstar_{i \in [n]} f_{\phi_i},
\]

where \( \bigstar \) denotes standard convolution operator over \( \mathbb{R} \) or over \( \mathbb{Z} \).
**Minimum of Random Variables.** Let $\Phi$ denote a random variable that equals to the minimum of $n$ i.i.d random variables $\{\phi_i\}_{i=1}^{n}$, i.e., $\Phi = \min_{i\in[n]} \phi_i$. Denote by $f_\phi(\cdot)$ and $F_\phi(\cdot)$ the p.d.f. and c.d.f. of $\phi \sim \phi_i$, respectively. Then, the p.d.f. and c.d.f. of $\Phi$ are given by

\[
\begin{align*}
    f_\Phi(x) &= n \cdot (1 - F_\phi(x))^{n-1} f_\phi(x), \quad \text{and} \\
    F_\Phi(x) &= 1 - (1 - F_\phi(x))^n.
\end{align*}
\]

(16) \hspace{1cm} (17)

**B.2 Computing Distributions of $X_l$ and $Y_l$**

The halting term of the recursion in Equations (7)-(9) is given by $Y_0$. Let $g_{\omega_l}(\cdot)$ denote the p.d.f. of the scaled random variable $\omega_l \gamma$, i.e.,

\[
g_{\omega_l}(y) = \frac{1}{\omega_l} f_{\gamma}(\frac{y}{\omega_l}).
\]

(18)

Then, the p.d.f. of $Y_0$ is simply written as

\[
f_{Y_0}(y) = g_{\omega_0}(y).
\]

(19)

In the case where $\gamma = 1 + \mathcal{N}(0, \sigma^2)$, Equation (19) simplifies to

\[
\begin{align*}
    f_{Y_0}(y) &= \frac{1}{\omega_0} f_{\mathcal{N}}(\frac{y}{\omega_0} - 1), \quad \text{and} \\
    F_{Y_0}(y) &= F_{\mathcal{N}}(\frac{y}{\omega_0} - 1).
\end{align*}
\]

(20) \hspace{1cm} (21)

Let $f^{*d}(\cdot)$ denote the $d$-fold convolution of a function $f(\cdot)$, i.e., the convolution of function $f(\cdot)$ with itself $d$ times. Following Equations (15)-(17), the recursion equations for the p.d.f.
and c.d.f. of $X_l$ and $Y_l$ is given by

$$f_{X_l}(x) = (d_R - 1)(1 - F_{Y_l}(x))^{d_R-2}f_{Y_l}(x),$$  \(\text{(22)}\)

$$F_{X_l}(x) = 1 - (1 - F_{Y_l}(x))^{d_R-1},$$  \(\text{(23)}\)

$$f_{Y_l}(y) = \left(g_{\omega_l} * f_{X_{l-1}}^{(d_R-1)}\right)(y), \quad \text{and}$$  \(\text{(24)}\)

$$F_{Y_l}(y) = \int_{-\infty}^{y} f_{Y_l}(t) dt.$$  \(\text{(25)}\)

Since we cannot analytically solve Equations (22)-(25), we use a numeric method based on quantization in order to represent and evaluate the functions $f_{X_l}(\cdot)$, $F_{X_l}(\cdot)$, $f_{Y_l}(\cdot)$, and $F_{Y_l}(\cdot)$. As suggested in [RU08], we compute a uniform sample of the functions, i.e., we consider the functions over the set $\delta\mathbb{Z}$, where $\delta$ denotes the quantization step size. Moreover, due to practical reasons we restrict the functions to a finite support, namely, $\{\delta k\}_{k=M}^{N}$ for some integers $M < N$. We denote the set $\{\delta k\}_{k=M}^{N}$ by $\delta[M,N]$. Obviously, the choice of $\delta$, $M$, and $N$ determines the precision of our target computation. Depending on the quantized function, it is also common to consider point masses at points not in $\delta[M,N]\mathbb{Z}$. For example, in case that the density function has an heavy tail above $\delta N$ we may assign the value $+\infty$ to the mass of the tail. The same applies analogously to a heavy tail below $\delta M$.

A Gaussian-like function (bell-shaped function) is bounded and continuous, and so are its derivatives. The area beneath its tails decays exponentially and becomes negligible a few standard deviations away from the mean. Thus, Gaussian-like functions are amenable to quantization. The parameters $M$ and $N$ are symmetric around the mean, and together with $\delta$ are chosen to make the error of a Riemann integral upper bounded by $e^{-(N-M)\delta/2}$. Therefore, we choose to zero the density functions outside the interval $[\delta M, \delta N]$. As we demonstrate by computations, the density functions $f_{X_l}(\cdot)$ and $f_{Y_l}(\cdot)$ are indeed bell-shaped, justifying the quantization. Figure 1 illustrates the p.d.f. of $X_0$ (Here $X_0$ equals to the minimum of $d_R - 1 = 5$ instances of $Y_0$). Note that by definition, $Y_0$ is a Gaussian random variable.

Computing $f_{Y_l}(\cdot)$ given $f_{X_{l-1}}(\cdot)$ requires convolution of functions. However, the restriction of the density functions to a restricted support $\delta[M,N]$ is not invariant under convolution. That
is, if the function $f$ is supported by $\delta[M, N]$, then $f \ast f$ is supported by $\delta[\frac{1}{2}(M + N) - (N - M), \frac{1}{2}(M + N) + (N - M)]$. In the quantized computations of $f_{X_l}(\cdot)$ and $f_{Y_l}(\cdot)$, our numeric calculations show that the mean and standard deviation of the random variables $X_l$ and $Y_l$ increase exponentially in $l$ as illustrated in Figures 2 and 3. Therefore, the slopes of the density functions $f_{X_l}(\cdot)$ and $f_{Y_l}(\cdot)$ decreases with $l$. This property allows us to double the quantization step $\delta$ as $l$ increases by one. Thus, the size of the support used for $f_{X_l}(\cdot)$ and $f_{Y_l}(\cdot)$ does not grow. Specifically, the interval $\delta[M, N]$ doubles but the doubling of $\delta$ keeps the number of points fixed. This method helps keep the computation tractable while keeping the error small.

For two quantized functions $f$ and $g$, the calculation of $f \ast g$ can be efficiently performed using Fast Fourier Transform (FFT). First, in order to prevent aliasing, extend the support with zeros (i.e., zero padding) so that it equals the support of $f \ast g$. Then, $f \ast g = \text{IFFT}(\text{FFT}(f) \times \text{FFT}(g))$ where $\times$ denotes a coordinate-wise multiplication. The outcome is scaled by the quantization step size $\delta$. In fact, the evaluation of $f_{Y_l}(\cdot)$ requires $d_L - 1$ convolutions and is performed in the frequency domain (without returning to the time domain in between) by a proper zero padding prior to performing the FFT.

Note that when $\gamma$ is a discrete random variable with a bounded support (as in [ADS09]), doubling applies to the demonstrated parameters, i.e. $d_L = 3$ and $\omega_l = 2^l$. 

\footnote{Doubling applies to the demonstrated parameters, i.e. $d_L = 3$ and $\omega_l = 2^l$.}
Figure 2: Probability density functions of $X_l$ for $l = 0, \ldots, 4$, $(d_L, d_R) = (3, 6)$ and $\sigma = 0.7$.

Figure 3: Probability density functions of $Y_l$ for $l = 0, \ldots, 4$, $(d_L, d_R) = (3, 6)$ and $\sigma = 0.7$. 
a precise computation of the probability distribution function of $X_s$ is obtained by following Equations (22)-(25).

### B.3 Estimating $\min_{t \geq 0} E e^{-tX_s}$

After obtaining a proper discretized representation of the p.d.f. of $X_s$ we approximate $E e^{-tX_s}$ for a given $t$ by

$$E e^{-tX_s} \approx \sum_{k=M}^{N} \delta \cdot f_{X_s}(\delta k) \cdot e^{-t\delta k}.$$ 

We then estimate the minimum value by searching over values of $t \geq 0$. Figure 4 depicts $\ln (E e^{-tX_s})$ as a function of $t \in (0, 0.5]$ for $s = 4, 6, 8, 10, 12$. The numeric calculations show that as $t$ grows from zero, the function $E e^{-tX_s}$ decreases to a minimum value, and then increases rapidly. We can also observe that both the values $\min_{t \geq 0} E e^{-tX_s}$ and $\arg \min_{t \geq 0} E e^{-tX_s}$ decreases as a function of $s$.

Following Lemma 19 we are interested in the maximum value of $\sigma$ for which Equation (13) holds for a given $s$. That is,

$$\sigma_0 \triangleq \sup \left\{ \sigma > 0 \mid \min_{t \geq 0} E e^{-tX_s} \cdot \left((d_R - 1)\frac{1}{2\sigma^2} \right)^{\frac{1}{\tau_2-2}} < 1 \right\}. \quad (26)$$

Note that if the set in Equation (26) is not empty, then it is an open interval $(0, \sigma_0) \in \mathbb{R}_+$. 
Figure 5: (a) Region for which $5e^{-\frac{1}{2\sigma^2}} E e^{-t X_4} < 1$ as a function of $t$ and $\sigma$ for $(d_L, d_R) = (3, 6)$. (b) Constant $c$ in Corollary~20 as a function of $\sigma$ in the case where $s = 4$ and $t = 0.11$, i.e., the value of $c$ over the cut of the $(t, \sigma)$-plane in plot (a) at $t = 0.11$ (depicted by a blue line).

Therefore, $\sigma_0$ is called a threshold. Figure 5 (a) illustrates the region in the $(t, \sigma)$ plane, for which Equation (13) holds with $s = 4$.

Let $t^*$ denote the value of $t$ that achieves the threshold $\sigma_0$. For every $\sigma \in (0, \sigma_0)$, we may set the value of the constant $c$ in Corollary 20 as

$$c = E e^{-t^* X_4} \cdot ((d_R - 1)e^{-\frac{1}{2\sigma^2}}) \frac{1}{\sigma^2}.$$

Figure 5 (b) illustrates the value of the constant $c$ in Corollary 20 as a function of $\sigma$ in the case where $s = 4$.

References


