On Decoding Irregular Tanner Codes

Guy Even [*] Nissim Halabi [†]

Abstract

We present a new combinatorial characterization for local-optimality of a codeword in irregular Tanner codes. This characterization is a generalization of [Arora, Daskalakis, Steurer; 2009] and [Vontobel; 2010]. The main novelty in this characterization is that it is based on a conical combination of subtrees in the computation trees. These subtrees may have any degree in the local-code nodes and may have any height (even greater than the girth). We prove that local-optimality in this new characterization implies Maximum-Likelihood (ML) optimality and LP-optimality. We also show that it is possible to compute efficiently a certificate for the local-optimality of a codeword given the channel output.

We apply this characterization to regular Tanner codes. We prove a lower bound on the noise threshold in channels such as BSC and AWGNC. When the noise is below this lower bound, the probability that LP decoding fails diminishes doubly exponentially in the girth of the Tanner graph.

We use local optimality also to design an iterative message-passing algorithm for decoding irregular LDPC codes. This new algorithm is guaranteed to find the locally optimal codeword if such a codeword exists. Moreover, an ML-certificate as well as an LP-certificate are proved if a locally optimal codeword exists.

[*] School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: guy@eng.tau.ac.il.

[†] School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: nissimh@eng.tau.ac.il.

1 Introduction

Modern coding theory deals with finding good error correcting codes that have efficient encoders and decoders ([RU08]). Many of the decoders for modern codes are suboptimal in the sense that they may fail to correct errors that are corrected by a maximum-likelihood (ML) decoder, but they are practical thanks to their simplicity and efficiency. Message-passing iterative decoding algorithms based on belief-propagation (see e.g., [Gal63, BGT93, Mac99, LMSS01, RU01]) and linear-programming (LP) decoding [Fel03, FWK05] are examples of such suboptimal decoders.

Many works deal with low-density parity-check (LDPC) codes and generalizations of LDPC codes. LDPC codes were first defined by Gallager [Gal63] who suggested several message-passing iterative decoding algorithms (e.g., "sum-product"). Tanner [Tan81] introduced graph representations of linear codes based on bipartite graphs over variable nodes and constraint nodes, and viewed iterative decoding as message-passing algorithms over the edges of the Tanner graph. In the standard setting, constraint nodes compute the parity function. In the generalized setting, constraint nodes use a local error-correcting code. One may view a constraint node with a linear local-code as a coalescing of multiple parity-check nodes. Therefore, a code may have a sparser and smaller representation when represented as a Tanner code in the generalized setting. Sipser and Spielman [SS96] studied Tanner codes based on expanders graphs and analyzed a simple bit-flipping decoding algorithm.

Wiberg *et al.* [WLK95, Wib96] developed the use of graphical models for systematically describing instances of known decoding algorithms. For example, the "sum-product" algorithm and the "min-sum" algorithm are generic iterative message-passing decoding algorithms that apply to any graph realization of a Tanner code. Wiberg *et al.* proved that the min-sum algorithm can be viewed as a dynamic programming algorithm that computes the ML-codeword if the Tanner graph is a tree. Although Tanner graphs are usually not trees, the min-sum algorithm proceeds as if the graph is a tree. For LDPC codes, Wiberg *et al.* characterized a necessary condition for decoding failures of the min-sum algorithm by "negative" cost trees, called *minimal deviations*.

Linear programming (LP) decoding was introduced by Feldman, Wainwright and Karger [Fel03, FWK05] for binary linear codes. LP-decoding is based on solving a fractional relaxation of an integer program that models the problem of ML-decoding. LP decoding has been applied to several codes, among them: RA codes, turbo-like codes, LDPC codes, and expander codes. Our work is motivated by the problem of finite-length and average-case analysis of successful LP-decoding of Tanner codes. There are very few works on this problem, and they deal only with specific cases. For example, Feldman and Stein [FS05] analyzed special expander codes, and Goldenberg and Burshtein [GB10] deal with repeat-accumulate codes.

Previous results. Combinatorial characterizations of sufficient conditions for successful decoding are based on so called "certificates". That is, given a received word y and a codeword x, we are interested in a one-sided error test that answers the questions: is x optimal with respect to y? is it unique? Note that the test may answer "no" for a positive instance. We call these tests *certificates* for the optimality of a codeword. Upper bounds on the word error probability are obtained by lower bounds on the probability that a certificate exists.

Koetter and Vontobel [KV06] analyzed LP decoding of regular LDPC codes. Their analysis is based on decomposing each codeword (and pseudocodeword) to a finite set of minimal structured trees (i.e., skinny trees) with uniform vertex weights. Arora *et al.* [ADS09] extended the work in [KV06] by introducing nonuniform weights to the vertices in the skinny trees, and defined *local-optimality*. For a BSC, Arora *et al.* proved that local optimality implies both ML-optimality and LP-optimality. They presented an analysis technique that performs a finite-length density evolution of a min-sum process to prove bounds on the probability of a decoding error. Arora *et al.* also pointed out that it is possible to design a re-weighted version of the min-sum decoder that finds the locally-optimal codeword if such exists. This work was further extended in [HE11] to memoryless channels. The analyses presented in these works [KV06, ADS09, HE11] are limited to skinny trees, the height of which is bounded by a quarter of the girth of the Tanner graph.

Vontobel [Von10] extended the decomposition of a codeword (and pseudocodeword) to skinny trees in graph covers (that originate in algebraic topology). This enabled Vontobel to mitigate the limitation on the height by the girth. The decomposition is obtained by a random walk, and applies also to irregular Tanner graphs.

Jian and Pfister [JP10] analyzed a special case of the attenuated max-product decoder [FK00], for regular LDPC codes. They considered skinny trees in the computation tree, the height of which is greater than the girth of the Tanner graph. Using contraction properties and consistency conditions, they proved sufficient conditions under which the message-passing decoder converges to a locally optimal codeword. This convergence also implies convergence to the LP-optimum and therefore to the ML-codeword.

Contributions. Our contribution is threefold. (i) We present a new combinatorial characterization of local-optimality for Tanner codes with respect to any memoryless binary-input output symmetric (MBIOS) channel. This characterization provides an ML-certificate and an LP-certificate for a given codeword. Based on this new characterization, we present two applications of local-optimality. (ii) In the case of regular Tanner codes, we present an analysis of LP-decoding failure. (iii) In the case of irregular LDPC codes, we present a new message passing decoding algorithm, called NWMS. The NWMS algorithm is guaranteed to find the locally optimal codeword if such exists. More details of our contributions are provided below.

A new combinatorial characterization of local-optimality for irregular Tanner codes with respect to any memoryless binary-input output-symmetric (MBIOS) channel is presented. This characterization uses subtrees in the computation tree in which the degree of local-code nodes is not limited to 2 (as opposed to skinny trees in previous analyses). We prove that local-optimality in this characterization implies ML-optimality (Theorem 5). We utilize the equivalence of graph cover decoding and LP-decoding for Tanner codes, implied by Vontobel and Koetter [VK05], to prove that local-optimality suffices also for LP-optimality (Theorem 7), as one would expect. We present an efficient dynamic programming algorithm that computes a local-optimality certificate for a codeword with respect to a given channel output.

Because trees in our new characterization may have degrees bigger than two, they contain more vertices. Hence this characterization leads to improved bounds for successful decoding of regular Tanner codes (Theorems 11 and 22). These bounds extend the probabilistic analysis of the min-sum process by Arora *et al.* [ADS09] to a sum-min-sum process on regular trees. For regular Tanner codes, we prove bounds on the word error probability of LP-decoding under MBIOS channels that are inverse doubly-exponential in the girth of the Tanner graph. We also prove bounds on the threshold of regular Tanner codes whose Tanner graphs have logarithmic girth. This means that if the noise in the channel is below that threshold, then the decoding error diminishes as a function of the block length. Note that Tanner graphs with logarithmic girth can be constructed explicitly (see e.g., [Gal63]).

Specifically, we consider as an example $(2, 16)$-regular Tanner codes with (i) $[16, 11, 4]$-extended Hamming codes as local-codes, and (ii) logarithmic girth Tanner graphs. The rate of such codes is at least 0.375. For the case of a binary symmetric channel (BSC) with bit flipping probability p, we prove a lower bound of $p^* = 0.044$ on the noise threshold. How does this result compare with results on expander Tanner codes? The error correction capability of expander codes depends on the expansion, thus a fairly large degree and huge block-lengths are required to achieve good error correction. Our example relies only on a 16-regular graph with logarithmic girth. Feldman and Stein [FS05] proved that LP decoding can asymptotically achieve capacity with a special family of expander Tanner codes. They also presented a worst-case analysis, which in the case of a code rate of 0.375, proves that LP decoding can recover any pattern of at most $0.0008N$ bit flips. This implies a lower bound of $p^* = 0.0008$ on the threshold. The best results for iterative decoding of such expander codes, reported by Skachek and Roth [SR03], imply a lower bound of $p^* = 0.0016$ on the threshold of a certain iterative decoder.

Finally, motivated by the weights and degree normalization in the characterization of local-optimality, we present a new message-passing iterative decoding algorithm for irregular LDPC codes, called the *normalized weighted min-sum* (NWMS) algorithm. The characterization of local-optimality for irregular LDPC codes has two parameters: (i) a certificate depth h, and (ii) a vector of layer weights $w \in \mathbb{R}_+^h$. We prove that the NWMS decoder computes the ML codeword if a locally-optimal codeword exists (Theorem 23). The time and message complexity of NWMS is $O(|E| \cdot h)$ where $|E|$ is the number of edges in the Tanner graph.

Various weighting methods of message-passing algorithms based on belief-propagation were explored by several researchers (see e.g., [FK00, CF02, CDE$^+$05, JP10]). The analyses and results of which are asymptotic (e.g., based on density evolution [RU01]) and limited to regular LDPC codes. Moreover, no bounds on the time and message complexity are proved. The NWMS algorithm comes with a guarantee for computing the ML codeword within h iterations if a local-optimality certificate of depth h exists for some codeword. Moreover, the output of NWMS can be efficiently certified. For the case of regular LDPC codes, the previous bounds on the probability that a local-optimality certificate exists [ADS09, HE11] also apply to the probability of NWMS decoding success.

The remainder of this paper is organized as follows. Section 2 provides background on ML-decoding and LP-decoding of Tanner codes over MBIOS channels. Section 3 presents combinatorial certificate, that applies both to ML-decoding and LP-decoding, for codewords of Tanner codes. In Section 4, we prove a structural decomposition for codewords of Tanner codes used as a key element in the proof of the main theorem of the previous section. In Section 5 we use the combinatorial characterization of local-optimality to bound the error probability of LP decoding for regular Tanner codes. Section 6 presents the NWMS iterative decoding algorithm for irregular LDPC codes, followed by a proof in Section 7 that NWMS finds the locally-optimal codeword if such exists. We conclude in Section 8.

2 Preliminaries

Graph Terminology. Let $\mathcal{N}_G(v)$ denote the set of neighbors of node v in graph G, and for a set $S \subseteq V$ let $\mathcal{N}_G(S) \triangleq \bigcup_{v\in S} \mathcal{N}_G(v)$. Let $P_{vu}(G)$ denote a shortest path between nodes v and u in G. Let $d_G(r,v)$ denote the distance (i.e., length of a shortest path) between nodes r and v in G, and let $girth(G)$ denote the length of the shortest cycle in G.

An *induced subgraph* is a subgraph obtained by deleting a set of vertices. The *subgraph of* $G = (V,E)$ *induced by* $S \subseteq V$, denoted by G_S, consists of S and all edges in E, both endpoints of which are contained in S.

Tanner-codes and Tanner graph representation. Let $G = (\mathcal{V} \cup \mathcal{J}, E)$ denote an edge-labeled bipartite-graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of N vertices called *variable nodes*, and $\mathcal{J} = \{C_1, \ldots, C_J\}$ is a set of J vertices called *local-code nodes*. We denote the degree of C_j by n_j.

Let $\overline{\mathcal{C}}^{\mathcal{J}} \triangleq \{\overline{\mathcal{C}}^j : \overline{\mathcal{C}}^j \text{ is an } [n_j,k_j,d_j] \text{ code}, \ j \in [J]\}$ denote a set of J *local-codes*. The local code $\overline{\mathcal{C}}^j$ corresponds to the local-code node $C_j \in \mathcal{J}$. We say that v_i *participates* in $\overline{\mathcal{C}}^j$ if (v_i, C_j) is an edge in E. The edges incident to each local-code node C_j are labeled $\{1, \ldots, n_j\}$. This labeling indicates the index of a variable nodes in the corresponding local-code.

Let a word $x = (x_1, \ldots, x_N) \in \{0,1\}^N$ denote an assignment to variable nodes in $\mathcal{V}$. Let $\mathcal{V}_j$ denote the ordered set of variable nodes in $\mathcal{N}_G(C_j)$ according to labels of edges incident to C_j. Denote by $x_{\mathcal{V}_j} \in \{0,1\}^{n_j}$ the projection of the word $x = (x_1, \ldots, x_N)$ onto entries associated with $\mathcal{V}_j$.

The *Tanner code* $\mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ based on the labeled *Tanner graph* G is the set of vectors $x \in \{0,1\}^N$ such that $x_{\mathcal{V}_j}$ is a codeword in $\overline{\mathcal{C}}^j$ for every $j \in [J]$.

Let d_j denote the minimum distance of the local code $\overline{\mathcal{C}}^j$. The *minimum local distance* d^* of a Tanner code $\mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ is the minimum distance of the local codes, i.e., $d^* = \min_j d_j$.

If the bipartite graph is (d_L, d_R)-regular, i.e., the vertices in $\mathcal{V}$ have degree d_L and the vertices in $\mathcal{J}$ have degree d_R, then the graph defines a (d_L, d_R)*-regular Tanner code*.

If the Tanner graph is sparse, i.e., $|E| = O(N)$, then it defines a *low-density Tanner code*. A *parity code* is the code that contains all binary words with even Hamming weight. Tanner codes with parity local codes that are based on sparse Tanner graphs are called *low-density parity-check (LDPC) codes*.

Consider a Tanner code $\mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}})$, where $\overline{\mathcal{C}}^{\mathcal{J}} = \{\overline{\mathcal{C}}^j\}_{j\in[J]}$. We say that a word $x = (x_1, ..., x_N)$ *satisfies* local-code $\overline{\mathcal{C}}^j$ if $x_{\mathcal{V}_j} \in \overline{\mathcal{C}}^j$. The set of words x that *satisfy* the local-code $\overline{\mathcal{C}}^j$ is denoted by $\mathcal{C}^j$, i.e., $\mathcal{C}^j = \{x \in \{0,1\}^N : x_{\mathcal{V}_j} \in \overline{\mathcal{C}}^j\}$. The resulting code $\mathcal{C}^j$ is the *extension* of the local-code $\overline{\mathcal{C}}^j$ from length n_j to length N. We denote the set of extended local-codes in $\overline{\mathcal{C}}^{\mathcal{J}}$ by $\mathcal{C}^{\mathcal{J}}$. Clearly, $\mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}}) \subseteq \mathcal{C}^j$. It holds that

$$\mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}}) = \bigcap_{j\in[J]} \mathcal{C}^j. \tag{1}$$

LP decoding of Tanner codes over memoryless channels. Let $c_i \in \{0,1\}$ and $y_i \in \mathbb{R}$ denote the ith transmitted binary symbol (channel input) and the ith received symbol (channel output), respectively. A *memoryless binary-input output-symmetric* (MBIOS) channel is defined by a conditional probability density function $f(y_i|c_i = a)$ for $a \in \{0,1\}$, that satisfies $f(y_i|0) = f(-y_i|1)$. The binary erasure channel (BEC), binary symmetric channel (BSC) and binary-input additive white Gaussian noise (BI-AWGN) channel are examples for MBIOS channels. In MBIOS channels, the *log-likelihood ratio* (LLR) vector $\lambda \in \mathbb{R}^N$ is defined by $\lambda_i(y_i) \triangleq \ln\left(\frac{f(y_i|c_i=0)}{f(y_i|c_i=1)}\right)$ for every input bit i. For a linear code $\mathcal{C}$, *Maximum-Likelihood (ML) decoding* is equivalent to

$$\hat{x}^{ML}(y) = \arg\min_{x \in \mathrm{conv}(\mathcal{C})} \langle \lambda(y), x \rangle, \tag{2}$$

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

In general, solving the optimization problem in (2) for linear codes is intractable [BMvT78]. Feldman *et al.* [Fel03, FWK05] introduced a linear programming relaxation for the problem of ML decoding of Tanner codes whose local codes are parity codes. This definition is based on a fundamental polytope that corresponds to the Tanner graph G. We consider an extension of this definition to the case in which the local codes are arbitrary as follows. The *generalized fundamental polytope* $\mathcal{P} \triangleq \mathcal{P}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ of a Tanner code $\mathcal{C} = \mathcal{C}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ is defined by

$$\mathcal{P} \triangleq \bigcap_{\mathcal{C}^j \in \overline{\mathcal{C}}^{\mathcal{J}}} \mathrm{conv}(\mathcal{C}^j). \tag{3}$$

Note that a Tanner code may have multiple representations by a Tanner graph and local codes. Moreover, different representations $(G, \overline{\mathcal{C}}^{\mathcal{J}})$ of the same Tanner code $\mathcal{C}$ may yield different generalized fundamental polytopes $\mathcal{P}(G, \overline{\mathcal{C}}^{\mathcal{J}})$. If the degree of each local-code node is constant, then the generalized fundamental polytope can be represented by $O(|\mathcal{J}|)$ variables and $O(|\mathcal{J}|)$ constraints. If, in addition, the Tanner graph is sparse, then $|\mathcal{J}| = O(N)$, and the generalized fundamental polytope has an efficient representation. Such Tanner codes are often called *generalized low-density parity-check codes*.

Given an LLR vector λ for a received word y, LP-decoding is defined by the following linear program:

$$\hat{x}^{LP}(y) \triangleq \arg\min_{x \in \mathcal{P}(G, \overline{\mathcal{C}}^{\mathcal{J}})} \langle \lambda(y), x \rangle. \tag{4}$$

The difference between ML-decoding and LP-decoding is that the fundamental polytope $\mathcal{P}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ may strictly contain the convex hull of $\mathcal{C}$. Vertices of $\mathcal{P}(G, \overline{\mathcal{C}}^{\mathcal{J}})$ that are not codewords of $\mathcal{C}$ must have fractional components and are called *pseudocodewords*.

3 A Combinatorial Certificate for an ML Codeword

In this section we present combinatorial certificates for codewords of Tanner codes that apply both to ML-decoding and LP-decoding. A certificate is a proof that a given codeword is the unique solution of maximum-likelihood decoding and linear-programming decoding. The certificate is based on combinatorial weighted structures in the Tanner graph, referred to as *local configurations*. These local configurations generalize the minimal configurations (skinny trees) presented by Vontobel [Von10] as extension to Arora *et al.* [ADS09]. We note that for Tanner codes, the support of each weighted local configuration is not necessarily a local valid configuration. For a given codeword, the certificate is computed by a dynamic-programming algorithm on the Tanner graph of the code.

Notation: Let $y \in \mathbb{R}^n$ denote the word received from the channel. Let $\lambda = \lambda(y)$ denote the LLR vector for y. Let $G = (\mathcal{V} \cup \mathcal{J}, E)$ denote a Tanner graph, and let $\mathcal{C}(G)$ denote a Tanner code based on G with minimum local distance d^*. Let $x \in \mathcal{C}(G)$ be a candidate for $\hat{x}^{ML}(y)$ and $\hat{x}^{LP}(y)$.

Definition 1 (Path-Prefix Tree). *Consider a graph $G = (V, E)$ and a node $r \in V$. Let $\hat{V}$ denote the set of all backtrackless paths in G with length at most h that start at node r, and let*

$$\hat{E} \triangleq \big\{(p_1, p_2) \in \hat{V} \times \hat{V} \mid p_1 \text{ is a prefix of } \mathrm{p}_2,\, |p_1| + 1 = |p_2|\big\}.$$

We identify the empty path in $\hat{V}$ with r. Denote by $\mathcal{T}_r^h(G) \triangleq (\hat{V}, \hat{E})$ the path-prefix tree *of G rooted at node r with height h. We denote the fact that a path $\hat{p} \in \hat{V}$ ends at $v \in V$, by $\hat{p} \sim v$.*

When dealing with the analysis of belief propagation algorithms on graphical models, the path-prefix tree of a Tanner graph G rooted at a variable node is usually referred to as the *computation tree*. We make the distinction between the computation tree and the path-prefix tree since we consider also path-prefix trees of subgraphs of a Tanner graph G and are not necessarily rooted at a variable node. We denote vertices in the path-prefix tree by $\hat{v}$,$\hat{u}$, etc. Vertices in G are denoted by v, u, etc.

The following definitions expand the combinatorial notion of minimal valid deviations [Wib96] and weighted minimal local-deviations (skinny trees) [ADS09, Von10] to the case of Tanner codes.

Definition 2 (d-tree). *Consider a Tanner graph $G = (\mathcal{V} \cup \mathcal{J}, E)$. A* d-tree, *$\mathcal{T}[r, h, d](G)$, of height h rooted at node r is a subtree of $\mathcal{T}_r^h(G)$ such that every variable node has full degree and every local-code node has degree d.*

Definition 3 (w-weighted subtree). *Consider a Tanner graph $G = (\mathcal{V} \cup \mathcal{J}, E)$. Let $\mathcal{T}_{\hat{r}} = (\hat{\mathcal{V}} \cup \hat{\mathcal{J}}, \hat{E})$ denote a subtree of $\mathcal{T}_r^h(G)$, and let $w = (w_1, \ldots, w_h) \in \mathbb{R}_+^h$ denote a non-negative weight vector. Let $\mathcal{T}_{\hat{r}}^{(w)} : \hat{\mathcal{V}} \backslash \{\hat{r}\} \to \mathbb{R}$ denote a weight function for variable nodes in $\mathcal{T}_{\hat{r}}$ as follows.*

$$\mathcal{T}_{\hat{r}}^{(w)}(\hat{v}) \triangleq \frac{w_t}{\deg_G(v)} \cdot \prod_{\hat{u} \in P_{\hat{r}\hat{v}} \backslash \{\hat{r}, \hat{v}\}} \frac{1}{\deg_{\mathcal{T}_{\hat{r}}}(\hat{u}) - 1}, \tag{5}$$

where $t = \lceil \frac{d(\hat{r}, \hat{v})}{2} \rceil$ and $\hat{v} \sim v$. Let $\mathcal{T}_{\hat{r}}^{(w)}$ denote the subtree $\mathcal{T}_{\hat{r}}$ with the weights defined in (5). We refer to $\mathcal{T}_{\hat{r}}^{(w)}$ as a w-weighted subtree.

For any w-weighted subtree $\mathcal{T}_{\hat{r}}^{(w)}$ of $\mathcal{T}_r^h(G)$, let $\pi_G[\mathcal{T}_{\hat{r}}^{(w)}] \in \mathbb{R}^{|\mathcal{V}|}$ denote the projection of $\mathcal{T}_{\hat{r}}^{(w)}$ to the Tanner graph G. That is, for every variable node v in G,

$$\pi_G[\mathcal{T}_{\hat{r}}^{(w)}](v) = \begin{cases} \sum_{\hat{v}: \hat{v} \sim v} \mathcal{T}_{\hat{r}}^{(w)}(\hat{v}) & \text{if } \{\hat{v} : \hat{v} \sim v\} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For two vectors $x \in \{0,1\}^N$ and $f \in [0,1]^N$, let $x \oplus f \in [0,1]^N$ denote the *relative point* defined by $(x \oplus f)_i = |x_i - f_i|$ [Fel03]. The following definition is an extension of local-optimality [ADS09, Von10] to Tanner codes on memoryless channels.

Definition 4 (local-optimality). *Let $\mathcal{C}(G) \subset \{0,1\}^N$ denote a Tanner code with minimum local distance d^*, and let $w \in [0,1]^h \backslash \{0^N\}$ denote a non-negative weight vector of length h. For any integer $2 \leqslant d \leqslant d^*$, let $\mathcal{B}_d^{(w)}$ denote the set of all vectors corresponding to projections by w-weighted d-trees to G, i.e., $\mathcal{B}_d^{(w)} = \big\{\pi_G[\mathcal{T}^{(w)}[r,2h,d](G)] \bigm| r \text{ is a variable node in } G\big\}$. A codeword $x \in \{0,1\}^N$ is (h,w,d)*-locally optimal for $\lambda \in \mathbb{R}^N$ *if for all vectors $\beta \in \mathcal{B}_d^{(w)}$,*

$$\langle \lambda, x \oplus \beta \rangle > \langle \lambda, x \rangle. \tag{7}$$

Note that $\mathcal{B}_d^{(w)} \subseteq [0,1]^N$ for every weight vector $w \in [0,1]^h$. Based on random walks on the Tanner graph, Vontobel showed that $(h,w,2)$-local optimality is sufficient both for ML-optimality and LP-optimality. The random walks are defined in terms derived from the generalized fundamental polytope. We extend the results of Vontobel [Von10] to “thicker” skinny-trees by using probabilistic combinatorial arguments on graphs and the properties of graph cover decoding [VK05]. Specifically, we prove that (h,w,d)-local optimality, for any $2 \leqslant d \leqslant d^*$, implies LP optimality (Theorem 7). Given the decomposition of Lemma 8 proved in Section 4, the following theorem is obtained by modification of the proof of [ADS09, Theorem 2] or [HE11, Theorem 6].

Theorem 5 (local-optimality is sufficient for ML). *Let $\mathcal{C}(G)$ denote a Tanner code with minimum local distance d^*. Let h be some positive integer and $w = (w_1, \ldots, w_h) \in [0,1]^h$ denote a non-negative weight vector. Let $\lambda \in \mathbb{R}^N$ denote the LLR vector received from the channel, and suppose that x is an (h,w,d)-locally optimal codeword for λ and some $2 \leqslant d \leqslant d^*$. Then x is also the unique maximum-likelihood codeword for λ.*

Proof. We use the decomposition proved in Section 4 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 8 there exists a distribution over the set $\mathcal{B}_d^{(w)}$, such that $\mathbb{E}_{\beta \in \mathcal{B}_d^{(w)}} \beta = \alpha z$. Let $f : [0,1]^N \to \mathbb{R}$ 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,

$$\begin{aligned}
\langle \lambda, x \rangle &< \mathbb{E}_{\beta \in \mathcal{B}_d^{(w)}} \langle \lambda, x \oplus \beta \rangle && \text{(by local-optimality of } x\text{)} \\
&= \langle \lambda, x \oplus \mathbb{E}\beta \rangle && \text{(by linearity of } f \text{ and linearity of expectation)} \\
&= \langle \lambda, x \oplus \alpha z \rangle && \text{(by Lemma 8)} \\
&= \langle \lambda, (1-\alpha)x + \alpha(x \oplus z) \rangle \\
&= \langle \lambda, (1-\alpha)x + \alpha x' \rangle \\
&= (1-\alpha)\langle \lambda, x \rangle + \alpha \langle \lambda, x' \rangle.
\end{aligned}$$

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 note that the characterization of graph cover decoding and its connection to LP decoding can be extended to the case of Tanner codes in the generalized setting. We use the terms and notation of Vontobel and Koetter [VK05] in the statement of Lemma 6. The following lemma shows that local-optimality based on d-trees is preserved after lifting to an M-cover. Note that the weight vector must be scaled by the cover degree M.

Lemma 6. *Let $\mathcal{C}(G)$ denote a Tanner code with minimum local distance d^*, and let $\tilde{G}$ denote any M-cover of G. Let $w \in [0, \frac{1}{M}]^h \backslash \{0^h\}$ for some positive integer h. Suppose that $x \in \mathcal{C}(G)$ is an (h, w, d)-locally optimal codeword for $\lambda \in \mathbb{R}^N$ for some $2 \leqslant d \leqslant d^*$. Let $\tilde{x} = x^{\uparrow M} \in \mathcal{C}(\tilde{G})$ and $\tilde{\lambda} = \lambda^{\uparrow M} \in \mathbb{R}^{N \cdot M}$ denote the M-lifts of x and λ, respectively. Then $\tilde{x}$ is an $(h, M \cdot w, d)$-locally optimal codeword for $\tilde{\lambda}$.*

Proof. Assume that $\tilde{x} = x^{\uparrow M}$ is not a $(h, M \cdot w, d)$-locally optimal codeword for $\tilde{\lambda} = \lambda^{\uparrow M}$. Then, there exists a d-tree $\mathcal{T} = \mathcal{T}[\tilde{r}, h, d](\tilde{G})$ rooted at some variable node $\tilde{r} \in \tilde{\mathcal{V}}$, such that the projection $\tilde{\beta} = \pi_{\tilde{G}}[\mathcal{T}^{(M \cdot w)}] \in [0, 1]^{N \cdot M}$ of the $(M \cdot w)$-weighted d-tree $\mathcal{T}^{(M \cdot w)}$ onto $\tilde{G}$ satisfies

$$\langle \tilde{\lambda}, \tilde{x} \oplus \tilde{\beta} \rangle \leqslant \langle \tilde{\lambda}, \tilde{x} \rangle. \tag{8}$$

Note that for $\tilde{x} \in \{0, 1\}^{N \cdot M}$ and its projection $x = p(\tilde{x}) \in \mathbb{R}^N$, it holds that

$$\frac{1}{M} \langle \tilde{\lambda}, \tilde{x} \rangle = \langle \lambda, x \rangle, \quad \text{and} \tag{9}$$

$$\frac{1}{M} \langle \tilde{\lambda}, \tilde{x} \oplus \tilde{\beta} \rangle = \langle \lambda, x \oplus \beta \rangle, \tag{10}$$

where $\beta = \pi_G[\mathcal{T}^{(w)}] \in [0, 1]^N$ is the projection of the w-weighted d-tree $\mathcal{T}$ onto the base graph G. From (8), (9), and (10) we get that $\langle \lambda, x \rangle \geqslant \langle \lambda, x \oplus \beta \rangle$, contradicting our assumption on the (h, w, d)-local optimality of x. Therefore, $\tilde{x}$ is a $(h, M \cdot w, d)$-locally optimal codeword for $\tilde{\lambda}$ in $\mathcal{C}(\tilde{G})$. □

The following theorem is obtained as a corollary of Theorem 5 and Lemma 6. The proof is based on arguments utilizing properties of graph cover decoding. Those arguments are used for a reduction from ML-optimality to LP-optimality similar to the reduction presented in the proof of [HE11, Theorem 8].

Theorem 7 (local optimality is sufficient for LP optimality)**.** *For every Tanner code $\mathcal{C}(G)$ with minimum local distance d^*, there exists a constant M such that, if*

1. $w \in [0, \frac{1}{M}]^h \backslash \{0^h\}$*, and*
2. x *is an* (h, w, d)*-locally optimal codeword for* $\lambda \in \mathbb{R}^N$ *and some* $2 \leqslant d \leqslant d^*$*,*

then x is also the unique optimal LP solution given λ.

3.1 Verifying local optimality

Let $G = (\mathcal{V} \cup \mathcal{J}, E)$ denote a Tanner graph, and let $\mathcal{C}(G)$ denote a Tanner code with minimum local distance d^*. Let h denote a positive integer and $w \in [0, 1]^h$. Consider a codeword $x \in \mathcal{C}(G)$ and any integer $2 \leqslant d \leqslant d^*$.

Let "$*$" denote a coordinate-wise vector multiplication. Lemma 29 implies that the mapping $(x, \lambda) \mapsto (0^N, b * \lambda)$, where $b_i = (-1)^{x_i}$, preserves local optimality. That is, verifying whether x is (h, w, d)-locally optimal for λ is equivalent to verifying that 0^N is (h, w, d)-locally optimal for $\lambda' \triangleq b * \lambda$. However, 0^N is locally optimal for λ' iff $\min_{\beta \in \mathcal{B}_d^{(w)}} \langle \lambda', \beta \rangle \geqslant 0$.

Given an LLR vector λ', one can find by a simple dynamic programming algorithm the w-weighted d-tree $\mathcal{T}^*$ rooted at variable node r, such that its projection β^* minimizes $\langle \lambda', \beta \rangle$

for all vectors β corresponding to projections of w-weighted d-trees rooted at r. Values are propagated from the leaves of $\mathcal{T}_r^{(w)}(G)$ to the root r. In every step, a node propagates to its parent the minimum cost of the sub d-tree that hangs from it in $\mathcal{T}_r^{(w)}(G)$, based on the minimum values received from its children. In fact, message-passing algorithms run dynamic programming algorithms on computation trees for every root in G simultaneously.

For a set S of real values, let $\min^{[i]}\{S\}$ denote the ith smallest member in S. Algorithm VERIFY-LO(x, λ, h, w, d), listed as Algorithm 1, is a message-passing algorithm that outputs *true* if a given codeword x is (h, w, d)-locally optimal for λ, otherwise returns *false*. For each edge (v, C), each iteration $l \in \{1, \ldots, h\}$ ("for" loop in Line 3) consists of one message $\mu_{v\to C}^{(l)}$ from the variable node v to the check node C, and one message $\mu_{C\to v}^{(l)}$ from C to v. Hence, the time and message complexity of Algorithm 1 is $O(|E| \cdot h)$.

Algorithm 1 VERIFY-LO(x, λ, h, w, d) - An iterative verification algorithm. Given an LLR vector $\lambda \in \mathbb{R}^{|\mathcal{V}|}$, a codeword $x \in \{0,1\}^{|\mathcal{V}|}$, level weights $w \in \mathbb{R}_+^h$, and parameter $d \in \mathbb{N}_+$, outputs "*true*" if x is (h, w, d)-locally optimal for λ, otherwise outputs "*false*".

```
 1: Initialize: $\forall v \in \mathcal{V} : \lambda'_v \leftarrow \lambda_v \cdot (-1)^{x_v}$
 2:             $\forall C \in \mathcal{J}, \forall v \in \mathcal{N}(C)$: $\mu_{C\to v}^{(-1)} \leftarrow 0$
 3: for $l = 0$ to $h - 1$ do
 4:     for all $v \in \mathcal{V}$, $C \in \mathcal{N}(v)$ do
 5:         $\mu_{v\to C}^{(l)} \leftarrow \frac{w_{h-l}}{\deg_G(v)}\lambda'_v + \frac{1}{\deg_G(v)-1}\sum_{C'\in\mathcal{N}(v)\setminus\{C\}} \mu_{C'\to v}^{(l-1)}$
 6:     end for
 7:     for all $C \in \mathcal{J}$, $v \in \mathcal{N}(C)$ do
 8:         $\mu_{C\to v}^{(l)} \leftarrow \frac{1}{d-1} \cdot \sum_{i=1}^{d-1} \min^{[i]} \left\{\mu_{v'\to C}^{(l)} \;:\; v' \in \mathcal{N}(C) \setminus \{v\}\right\}$
 9:     end for
10: end for
11: for all $v \in \mathcal{V}$ do
12:     $\mu_v \leftarrow \sum_{C\in\mathcal{N}(v)} \mu_{C\to v}^{(h-1)}$
13:     if $\mu_v \leqslant 0$ then {min-cost $w$-weighted $d$-tree rooted at $v$ has non-positive value}
14:         return false;
15:     end if
16: end for
17: return true;
```

4 Constructing Codewords from Weighted Trees Projections

This section features Lemma 8, which is the key structural lemma in the proof of Theorem 5. This Lemma shows that every codeword of a Tanner code can be constructed by a summation over a finite set of projections of weighted trees in the computation trees of G.

Lemma 8. *Let $\mathcal{C}(G)$ denote a Tanner code with minimum local distance d^*, and let h denote some positive integer. For every codeword $x \neq 0^N$, and for every $2 \leqslant d \leqslant d^*$, there exists a distribution over d-trees $\mathcal{T}$ of G of height h and a positive integer H such that, for every weight vector $w \in [0, \frac{1}{H}]^h \backslash \{0^h\}$, there exists an $\alpha \in (0, 1]$, such that*

$$\mathbb{E}_{\mathcal{T}\in\mathcal{B}_d^{(w)}}\big[\pi_G[\mathcal{T}]\big] = \alpha x.$$

Proof sketch. Every codeword $x \in \mathcal{C}(G)$ can be decomposed into $\|x\|_1$ weighted path-prefix trees (see Lemma 9). Every weighted path-prefix tree is a convex combination of weighted d-trees (see Lemma 10). Putting these two results together yields Lemma 8. $\square$

For a codeword $x \in \mathcal{C}(G) \subset \{0,1\}^N$, let G_x denote the subgraph of the Tanner graph G induced by $V_x \cup \mathcal{N}(V_x)$ where $V_x = \{v_i \mid x_i = 1\}$.

Lemma 9. *Let $\mathcal{C}(G)$ denote a Tanner code and let h denote some positive integer. For every codeword $x \neq 0^N$, and for every weight vector $w \in \mathbb{R}_+^h$,*

$$\big(\sum_{t=1}^{h} w_t\big) \cdot x = \sum_{r : x_r = 1} \pi_G[\mathcal{T}_r^{(w)}(G_x)].$$

Proof. Let us consider two variable nodes $u, v \in G_x$. Notice that $|\{\hat{v} \in \mathcal{T}_u^h(G_x) \ : \ \hat{v} \sim v\}| = |\{\hat{u} \in \mathcal{T}_v^h(G_x) \ : \ \hat{u} \sim u\}|$. Indeed, for every path from the root of $\mathcal{T}_u^h(G_x)$ to a node $\hat{v} \in \{\hat{v} : \hat{v} \sim v\}$, there exists a unique reversed path in $\mathcal{T}_v^h(G_x)$ from the root to a node $\hat{u}$ such that $\hat{u} \sim u$. Let $\overrightarrow{p} = (v, \ldots, \hat{r})$ denote a path in the path-prefix tree $\mathcal{T}_v^h$ rooted at v, then $\overleftarrow{p} = (r, \ldots, \hat{v})$ denotes the corresponding reversed path in the path-prefix tree $\mathcal{T}_r^h$.

Consider an all-one weight vector $\eta = 1^h$. In (11)-(12), let $\mathcal{T}_r^{(\eta)} \triangleq \mathcal{T}_r^{(\eta)}(G_x)$, $\deg(\cdot) \triangleq \deg_{G_x}(\cdot)$, $d(\cdot,\cdot) \triangleq d_{\mathcal{T}_v^{2h}(G_x)}(\cdot,\cdot)$, $\hat{r} \sim r$, and $\hat{u} \sim u$. Let $q \circ p$ denote the concatenation of path q with path p. Equation (11) holds for every $1 \leqslant i \leqslant 2h$.

$$\begin{aligned}
\sum_{\{\overrightarrow{p}=(v,\ldots,\hat{r}) : d(v,\hat{r})=i\}} \mathcal{T}_r^{(\eta)}(\overleftarrow{p}) &= \sum_{\{\overrightarrow{q}=(v,\ldots,\hat{u}) : d(v,\hat{u})=i-1\}} \ \sum_{\{\hat{r} \in \mathcal{N}(\hat{u}) : d(v,\hat{r})=i\}} \mathcal{T}_r^{(\eta)}\big(\overleftarrow{\overrightarrow{q} \circ (r)}\big) \\
&= \sum_{\{\overrightarrow{q}=(v,\ldots,\hat{u}) : d(v,\hat{u})=i-1\}} \ \sum_{\{\hat{r} \in \mathcal{N}(\hat{u}) : d(v,\hat{r})=i\}} \frac{1}{\deg(u)-1} \mathcal{T}_u^{(\eta)}(\overleftarrow{q}) \\
&= \sum_{\{\overrightarrow{q}=(v,\ldots,\hat{u}) : d(v,\hat{u})=i-1\}} \mathcal{T}_u^{(\eta)}(\overleftarrow{q}) \cdot \sum_{\{\hat{r} \in \mathcal{N}(\hat{u}) : d(v,\hat{r})=i\}} \frac{1}{\deg(u)-1} \\
&= \sum_{\{\overrightarrow{q}=(v,\ldots,\hat{u}) : d(v,\hat{u})=i-1\}} \mathcal{T}_u^{(\eta)}(\overleftarrow{q}). \qquad (11)
\end{aligned}$$

Note that the reversed paths $\overleftarrow{p}$ and $\overleftarrow{q}$ in the summations of (11) end at a node $\hat{v}$ such that $\hat{v} \sim v$. Equation (11) implies that the sum of all η-weighted assignments to nodes $\hat{v} \sim v$ in $\{\mathcal{T}_r^{(\eta)}(G_x) \ : \ x_r = 1\}$ that correspond to paths of length i does not depend on i.

In particular, for $i = 1$, $\sum_{\{\overrightarrow{p}=(v,\hat{r})\}} \mathcal{T}_r^{(\eta)}(\overleftarrow{p}) = 1$. It follows that for every $1 \leqslant i \leqslant 2h$,

$$\sum_{\{\overrightarrow{p}=(v,\ldots,\hat{r}) : d(v,\hat{r})=i\}} \mathcal{T}_r^{(\eta)}(\overleftarrow{p}) = 1. \qquad (12)$$

Note that for every two variable nodes v, r, it holds that $\mathcal{T}_r^{(w)}(\hat{v}) = w_{d(r,\hat{v})/2} \cdot \mathcal{T}_r^{(\eta)}(\hat{v})$. Hence, $\sum_{\{\overrightarrow{p}=(v,\ldots,\hat{r}) : d(v,\hat{r})=2i\}} \mathcal{T}_r^{(w)}(\overleftarrow{p}) = w_i$. We conclude that for every variable node v in G_x

$$\sum_{r : x_r = 1} \pi[\mathcal{T}_r^{(w)}(G_x)](v) = \big(\sum_{i=1}^{h} w_i\big), \qquad (13)$$

and the claim follows. $\square$

Lemma 10. *For every connected subgraph G_S of a Tanner graph G, let d^* denote the minimal degree of a local-code node in G_S. Then for every variable node $r \in G_S$, a positive integer h, $2 \leqslant d \leqslant d^*$, and every weight vector $w \in \mathbb{R}^h_+$, it holds that*

$$\mathcal{T}_r^{(w)}(G_S) = \mathbb{E}\big[\mathcal{T}^{(w)}[r, 2h, d](G_S)\big]$$

with respect to a uniform distribution over d-trees $\mathcal{T}$ of G_S rooted at r with height $2h$.

Proof. Consider a subgraph G_S of a Tanner graph G, and a positive integer $d \leqslant d^*$. Let $\mathcal{T}_r^{(w)}(G_S)$ denote an w-weighted path-prefix tree rooted at node r with height $2h$. We want to show that the uniform distribution over w-weighted d-trees has the property that the expectation of trees over the distribution equals $\mathcal{T}_r^{(w)}(G_S)$.

We grow a d-tree rooted at r randomly in the path-prefix tree $\mathcal{T}_r^{2h}(G_S)$. That is, start from the root r. For each variable node take all it's children, and for each local-code node choose d distinct children uniformly at random. Let $\mathcal{T}[r, 2h, d]$ denote such a random d-tree, and consider a variable node $\hat{v} \in \mathcal{T}_r^{2h}(G_S)$. Note that $\mathcal{T}^{(w)}[r, 2h, d](\hat{v})$ is constant and does not depend on the random process. Equation (14) develops the equality

$$\mathbb{E}\big[\mathcal{T}^{(w)}[r, 2h, d](\hat{v})\big] = \mathcal{T}_r^{(w)}(\hat{v}).$$

$$\begin{aligned}
\mathbb{E}\big[\mathcal{T}^{(w)}[r, 2h, d](\hat{v})\big] &= \sum_{\{\mathcal{T}[r,2h,d] \in \mathcal{T}_r^{2h}(G_S)\}} \Pr(\mathcal{T}[r, 2h, d]) \cdot \mathcal{T}^{(w)}[r, 2h, d](\hat{v}) \\
&= \sum_{\{\mathcal{T}[r,2h,d] \in \mathcal{T}_r^{2h}(G_S) : \hat{v} \in \mathcal{T}[r,2h,d]\}} \Pr(\mathcal{T}[r, 2h, d]) \cdot \mathcal{T}^{(w)}[r, 2h, d](\hat{v}) \\
&= \mathcal{T}^{(w)}[r, 2h, d](\hat{v}) \cdot \sum_{\{\mathcal{T}[r,2h,d] \in \mathcal{T}_r^{2h}(G_S) : \hat{v} \in \mathcal{T}[r,2h,d]\}} \Pr(\mathcal{T}[r, 2h, d]) \\
&= \mathcal{T}^{(w)}[r, 2h, d](\hat{v}) \cdot \Pr(\hat{v} \in \mathcal{T}[r, 2h, d]) \\
&= \mathcal{T}^{(w)}[r, 2h, d](\hat{v}) \cdot \prod_{\hat{u} \in P_{r\hat{v}} \setminus \{r, \hat{v}\} \cap \hat{\mathcal{J}}} \frac{d-1}{\deg(\hat{u}) - 1} \\
&= \frac{w_{d(r,\hat{v})/2}}{\deg(\hat{v}) \cdot (d-1)^{d(\hat{r},\hat{v})/2}} \cdot \prod_{\hat{u} \in P_{r\hat{v}} \setminus \{r, \hat{v}\} \cap \hat{\mathcal{V}}} \frac{1}{\deg(\hat{u}) - 1} \\
&\quad \cdot \prod_{\hat{u} \in P_{r\hat{v}} \setminus \{r, \hat{v}\} \cap \hat{\mathcal{J}}} \frac{d-1}{\deg(\hat{u}) - 1} \\
&= \frac{w_{d(r,\hat{v})/2}}{\deg(\hat{v})} \cdot \prod_{\hat{u} \in P_{r\hat{v}} \setminus \{r\}} \frac{1}{\deg(\hat{u}) - 1} \\
&= \mathcal{T}_r^{(w)}(\hat{v}) \qquad (14)
\end{aligned}$$

as required. □

5 Bounds on Error Probability Using Local-Optimality

In this section we analyze the probability that a local optimality certificate for regular Tanner codes exists, and therefore LP decoding succeeds. The analysis is based on the study of a "sum-min-sum" process that characterizes d-trees of a regular Tanner graph. We prove upper bounds on the error probability of LP decoding of regular Tanner codes in memoryless channels. The upper bounds on the error probability imply lower bounds on the threshold of LP decoding. We apply the analysis to binary symmetric channels, and compare our results with previous results on expander codes. The analysis presented in this section generalizes the probabilistic analysis of Arora *et al.* [ADS09] from 2-trees (skinny trees) to d-trees for any $d \geqslant 2$.

In the remainder of this section, we restrict our discussion to (d_L, d_R)-regular Tanner codes with minimum local distance d^* among the local codes. Let d denote a parameter such that $2 \leqslant d \leqslant d^*$.

Theorem 11 summarizes the main results presented in this section for binary symmetric channels, and generalizes to any MBIOS channel as described in Section 5.3. Concrete bounds are given for a $(2, 16)$-regular Tanner code with code rate at least 0.375 when using $[16, 11, 4]$-extended Hamming codes as local codes.

Theorem 11. *Let G denote a (d_L, d_R)-regular bipartite graph with girth g, and let $\mathcal{C}(G)$ denote a Tanner code based on G with minimum local distance d^* of the local codes. Let $x \in \mathcal{C}(G)$ be a codeword. Suppose that $y \in \{0,1\}^N$ is obtained from x by flipping every bit independently with probability p. Then,*

1. *[finite length bound] Let $d = d_0$, $p \leqslant p_0$, $(d_L, d_R) = (2, 16)$, and $d^* = 4$. For the values of d_0 and p_0 in Table 1a it holds that x is the unique optimal solution to the LP decoder with probability at least*

$$\Pr\big(LP(y) = x\big) \geqslant 1 - N \cdot c^{(d-1)^{\lfloor \frac{1}{4} g \rfloor}}$$

   for some constant $c < 1$.

2. *[asymptotic bound] Let $d = d_0$, $(d_L, d_R) = (2, 16)$, $d^* = 4$, and $g = \Omega(\log N)$ sufficiently large. For the values of d_0 and p_0 in Table 1b it holds that x is the unique optimal solution to the LP decoder with probability at least $1 - exp(-N^{\gamma})$ for some constant $0 < \gamma < 1$, provided that $p \leqslant p_0(d_0)$.*

3. *For any (d_L, d_R) and $2 \leqslant d \leqslant d^*$ s.t. $(d_L - 1)(d - 1) \geqslant 2$, the codeword x is the unique optimal solution to the LP decoder with probability at least $1 - N \cdot c^{((d_L-1)(d-1))^{\lfloor \frac{1}{4} g \rfloor}}$ for some constant $c < 1$, provided that*

$$\min_{t \geqslant 0} \left\{ \big(c_1(p, d, d_L, d_R, t)\big) \cdot \big(c_2(p, d, d_L, d_R, t)\big)^{1/(d'_L \cdot d' - 1)} \right\} < 1,$$

   where

$$\begin{aligned} c_1(p, d, d_L, d_R, t) &= \sum_{k=0}^{d'-1} \binom{d'_R}{k} p^k (1-p)^{(d'_R - k)} e^{-t(d'-2k)} + \left(\sum_{k=d'}^{d'_R} \binom{d'_R}{k} p^k (1-p)^{d'_R - k} \right) e^{td'}, \\ c_2(p, d, d_L, d_R, t) &= \binom{d'_R}{d'} \big((1-p)e^{-t} + pe^{t}\big)^{d'}. \end{aligned}$$

	d_0	p_0
“finite”	3	0.0086
	4	0.0218
“asymptotic”	3	0.019
	4	0.044

Table 1: Computed values of p_0 for finite $d_0 < d^*$ in Theorem 11. Values are presented for $(2, 16)$-Tanner code with rate at least 0.375 when using $[16, 11, 4]$-extended Hamming codes as local codes. (a) finite-length bound: $\forall p \leqslant p_0$ bound on the word error probability that is inverse doubly-exponential in the girth of the Tanner graph. (b) asymptotic-bound: For $g = \Omega(\log N)$ sufficiently large, LP decoder succeeds w.p. at least $1 - exp(-N^{\gamma})$ for some constant $0 < \gamma < 1$, provided that $p \leqslant p_0(d_0)$.

Proof Outline. Theorem 11 follows from Lemma 14, Lemma 17, Corollary 20, and Corollary 21 as follows. The first part, that states a finite-length result, follows from Lemma 14 and Corollaries 20 and 21 by taking $s = 0 < h < \frac{1}{4} girth(G)$ which holds for any Tanner graph G. The second part, that deals with an asymptotic result, follows from Lemma 14 and Corollaries 20 and 21 by fixing $s = 10$ and taking $g = \Omega(\log N)$ sufficiently large such that $s < h = \Theta(\log N) < \frac{1}{4} girth(G)$. It therefore provides a lower bound on the threshold of LP-decoding. The third part, that states a finite-length result for any (d_L, d_R)-regular LDPC code, follows from Lemma 14 and Lemma 17. □

How does this result compare with results on expander Tanner codes? The error correction capability of expander codes depends on the expansion, thus a fairly large degree and huge block-lengths are required to achieve good error correction. Our example for which results are stated in Theorem 11.1 and 11.2 relies only on a 16-regular graph with logarithmic girth. Sipser and Spielman [SS96] studied Tanner codes based on expanders graphs and analyzed a simple bit-flipping iterative decoding algorithm. Their novel scheme was further improved over the years to follow, and it was shown that expander Tanner codes can *asymptotically* achieve capacity in the BSC with iterative decoding bit-flipping scheme [Z0́1, BZ02, BZ04]. In these works, a worst-case analysis was performed as well. The best result for iterative decoding of such expander codes, reported by Skachek and Roth [SR03], implies a lower bound of $p^* = 0.0016$ on the threshold of a certain iterative decoder for rate 0.375 codes. Feldman and Stein [FS05] proved that LP-decoding can *asymptotically* achieve capacity with a special family of expander Tanner codes. They also presented a worst-case analysis, which in the case of a code rate of 0.375, proves that LP decoding can recover any pattern of at most $0.0008N$ bit flips. This implies a lower bound of $p^* = 0.0008$ on the threshold. Those analyses yield overly pessimistic predictions for the average-case. Theorem 11.2 implies that LP-decoding can correct up to $0.044N$ bit flips with high probability.

We now provide more details and prove the lemmas and corollaries used in the proof of Theorem 11.

In order to simplify the probabilistic analysis of algorithms for decoding linear codes over symmetric channels, we apply the assumption that the all-zero codeword was transmitted, i.e., $c = 0^N$. Note that the correctness of the all-zero assumption depends on the employed decoding algorithm. Although this assumption is trivial for ML decoding because of the symmetry of a linear code $\mathcal{C}(G)$, it is not immediately clear in the context of LP-decoding. Feldman *et al.* [Fel03, FWK05] noticed that the fundamental polytope $\mathcal{P}(G)$ of Tanner codes with parity-check local codes is highly symmetric, and proved that for binary-input output-symmetric channels, the probability that the LP decoder fails is independent of the transmitted codeword. The symmetry property of the polytope remains also for the generalized fundamental polytope of Tanner codes based on non-trivial linear local codes. Therefore, one can assume that $c = 0^N$ when analyzing LP-decoding failure for linear Tanner codes. The following lemma gives a structural characterization for the event of LP-decoding failure if $c = 0^N$.

Lemma 12. *Assume that the all-zero 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 to the all-zero codeword, then for every $w \in \mathbb{R}_+^h \backslash \{0^h\}$ and $2 \leqslant d \leqslant d^*$ there exists a vector $\beta \in \mathcal{B}_d^{(w)}$ such that $\langle \lambda, \beta \rangle \leqslant 0$.*

Proof. Consider the event where the LP decoder fails to decode the all-zero codeword, i.e., 0^N is not a unique optimal LP solution. Theorem 7 implies that there exists a constant M such that, for every $w' \in [0, \frac{1}{M}]^h \backslash \{0^h\}$, the all-zero codeword is not (h, w', d)-locally optimal for λ. That is, there exists a vector $\beta' = \pi_G[\mathcal{T}^{(w')}[r, 2h, d](G)] \in \mathcal{B}_d^{(w')}$ such that $\langle \lambda, \beta' \rangle \leqslant 0$. The lemma follows by assigning $w' = \frac{1}{M \cdot ||w||_\infty} \cdot w$, and scaling β' by $M \cdot ||w||_\infty$ to obtain β, as required. □

We therefore have for a fixed h and $w \in \mathbb{R}_+^h \backslash \{0^h\}$ that

$$\Pr\{\text{LP decoding fails}\} \leqslant \Pr\left\{\exists \beta \in \mathcal{B}_d^{(w)} \text{ such that } \langle \lambda, \beta \rangle \leqslant 0 \middle| c = 0^N\right\}. \qquad (15)$$

5.1 Bounding Processes on Trees

Let G be a (d_L, d_R)-regular Tanner graph, and fix $h < \frac{1}{4} girth(G)$. Let $\mathcal{T}_{v_0}^{2h}(G)$ denote the path-prefix tree rooted at a variable node v_0 with height $2h$. Since $h < \frac{1}{4} girth(G)$, it follows that the projection of $\mathcal{T}_{v_0}^{2h}(G)$ to G is a tree. We direct the edges of $\mathcal{T}_{v_0}$ so that it is an in-branching directed toward the root v_0 (i.e., a rooted spanning tree with directed paths to the root v_0 from all the nodes). For $l \in \{0, \ldots, 2h\}$, denote by V_l the set of vertices of $\mathcal{T}_{v_0}^{2h}$ at height l (the leaves have height 0 and the root has height $2h$). Let $\tau \subseteq V(\mathcal{T}_{v_0}^{2h})$ denote the vertex set of a d-tree rooted at v_0.

Definition 13 ((h, ω, d)-Process on a (d_L, d_R)-Tree)**.** *Let $\omega \in \mathbb{R}_+^h$ denote a weight vector. Let λ denote an assignment of real values to the variable nodes of $\mathcal{T}_{v_0}$, we define the ω-weighted value of a d-tree τ by*

$$val_\omega(\tau; \lambda) \triangleq \sum_{l=0}^{h-1} \sum_{v \in \tau \cap V_{2l}} \omega_l \cdot \lambda_v.$$

Namely, the sum of the values of variable nodes in τ weighted according to their height.

Given a probability distribution over assignments λ, we are interested in the probability

$$\Pi_{\lambda, d, d_L, d_R}(h, \omega) \triangleq \Pr_\lambda \left\{ \min_{\tau \subset \mathcal{T}_{v_0}^{2h}:\, \tau\ d\text{−}tree} val_\omega(\tau; \lambda) \leqslant 0 \right\}. \qquad (16)$$

In other words, $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$ is the probability that the minimum value over all d-trees of height $2h$ 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 $\mathcal{T}_{v_0}^{2h}$ and $\mathcal{T}_{v_1}^{2h}$ are isomorphic, hence $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$ does not depend on the root v_0.

With this notation, the following lemma connects between the (h,ω,d)-process on (d_L,d_R)-trees and the event where the all-zero codeword is (h,w,d)-locally optimal. We apply a union bound utilizing Lemma 12, 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 $h < \frac{1}{4}girth(G)$. Suppose that $\lambda \in \mathbb{R}^N$ is the log-likelihood ratio of the word received from the channel. Then, the transmitted codeword $c = 0^N$ is $(h,\alpha\cdot w,d)$-locally optimal for $\alpha \triangleq (M\cdot||w||_\infty)^{-1}$ with probability at least*

$$1 - N\cdot\Pi_{\lambda,d,d_L,d_R}(h,\omega),\quad \text{where } \omega_l = w_{h-l}\cdot d_L^{-1}\cdot(d_L-1)^{l-h+1}\cdot(d-1)^{h-l},$$

and with at least the same probability, $c = 0^N$ is also the unique optimal LP solution given λ.

Note the two different weight notations that we use for consistency with [ADS09]: (i) w denotes weight vector in the context of (h,w,d)-local optimality certificate, and (ii) ω denotes weight vector in the context of d-trees in the (h,ω,d)-process. A one-to-one correspondence between these two vectors is given by $\omega_l = w_{h-l}\cdot d_L^{-1}\cdot(d_L-1)^{l-h+1}\cdot(d-1)^{h-l}$ for $0\leqslant l<T$. From this point on, we will use only ω in this section.

Following Lemma 14, it is sufficient to estimate the probability $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$ for a given weight vector ω, a distribution of a random vector λ, constant $2\leqslant d\leqslant d^*$,and degrees (d_L,d_R). Arora *et al.* [ADS09] introduced a recursion for estimating and bounding the probability of the existence of a 2-tree (skinny tree) with non-positive value in a $(h,\omega,2)$-process. We generalize the recursion and its analysis to d-trees with $2\leqslant d\leqslant d^*$.

For a set S of real values, let $\min^{[i]}\{S\}$ denote the ith smallest member in S. Let $\{\gamma\}$ denote an ensemble of i.i.d. random variables. Define random variables $X_0,\ldots,X_{h-1}$ and $Y_0,\ldots,Y_{h-1}$ with the following recursion:

$$Y_0 = \omega_0\gamma \tag{17}$$

$$X_l = \sum_{i=1}^{d-1}\min{}^{[i]}\{Y_l^{(1)},\ldots,Y_l^{(d_R-1)}\}\qquad(0\leqslant l<h) \tag{18}$$

$$Y_l = \omega_l\gamma + X_{l-1}^{(1)}+\ldots+X_{l-1}^{(d_L-1)}\qquad(0<l<h) \tag{19}$$

The notation $X^{(1)},\ldots,X^{(k)}$ and $Y^{(1)},\ldots,Y^{(k)}$ denotes k mutually independent copies of the random variables X and Y, respectively. Each instance of Y_l, $0\leqslant l<h$, uses an independent instance of a random variable γ. Note that for every $0\leqslant l<h$, the $d-1$ order statistic random variables $\big\{\min^{[i]}\{Y_l^{(1)},\ldots,Y_l^{(d_R-1)}\}\ :\ 1\leqslant i\leqslant d-1\big\}$ in Equation (18) are dependent.

Consider a directed tree $\mathcal{T}=\mathcal{T}_{v_0}$ of height $2h$, 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 $0\leqslant l<h$. Note that any realization of the random variables $\{\gamma\}$ to variable nodes in $\mathcal{T}$ can be viewed as an assignment λ. Thus, the minimum value of a d-tree of $\mathcal{T}$ equals $\sum_{i=1}^{d_L}X_{h-1}^{(i)}$. This implies that the recursion in (17)-(19) defines a dynamic programming algorithm for computing

$\min_{\tau\subset\mathcal{T}:\ \tau\ d-tree}\, val_\omega(\tau;\lambda)$. Now, let the components of the LLR vector λ be i.i.d. random variables distributed identically to $\{\gamma\}$, then

$$\Pi_{\lambda,d,d_L,d_R}(h,\omega) = \Pr\left\{\sum_{i=1}^{d_L} X_{h-1}^{(i)} \leqslant 0\right\}. \tag{20}$$

Given a distribution of $\{\gamma\}$ and a finite "height" h, it is possible to compute the distribution of X_l and Y_l according to the recursion in (17)-(19). The following two lemmas play a major role in proving bounds on $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$.

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

$$\Pi_{\lambda,d,d_L,d_R}(h,\omega) \leqslant \left(\mathbb{E}e^{-tX_{h-1}}\right)^{d_L}.$$

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

Lemma 16 (following [ADS09]). *For $0 \leqslant s < l < h$, we have*

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

Proof. We prove the claim by induction on the difference $l-s$. We first derive an equality for $\mathbb{E}e^{-tY_l}$ and a bound for $\mathbb{E}e^{-tX_l}$. Since Y_l is the sum of mutually independent variables,

$$\mathbb{E}e^{-tY_l} = \left(\mathbb{E}e^{-t\omega_l\gamma}\right)\left(\mathbb{E}e^{-tX_{l-1}}\right)^{d'_L}. \tag{21}$$

By definition of X_l we have the following bound,

$$\begin{aligned} e^{-tX_l} &= e^{-t\sum_{j=1}^{d'}\min^{[j]}\{Y_l^{(i)}:1\leqslant i\leqslant d'_R\}} \\ &= \prod_{j=1}^{d'} e^{-t\min^{[j]}\{Y_l^{(i)}:1\leqslant i\leqslant d'_R\}} \\ &\leqslant \sum_{S\subseteq[d'_R]:|S|=d'}\ \prod_{i\in S} e^{-tY_l^{(i)}}. \end{aligned}$$

Therefore, from linearity of expectation and since $\{Y_l^{(i)}\}_{i=1}^{d'_R}$ are mutually independent variables, we have

$$\mathbb{E}e^{-tX_l} \leqslant \binom{d'_R}{d'}\left(\mathbb{E}e^{-tY_l}\right)^{d'}. \tag{22}$$

By substituting (21) in (22), we get

$$\mathbb{E}e^{-tX_l} \leqslant \left(\mathbb{E}e^{-tX_{l-1}}\right)^{(d'_L\cdot d')}\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\omega_l\gamma}\right)^{d'}, \tag{23}$$

which proves the induction basis where $s = l-1$. Suppose, therefore, that the lemma holds for $l-s=i$, we now prove it for $l-(s-1)=i+1$. Then by substituting (23) in the induction hypothesis, we have

$$\begin{aligned}
\mathbb{E}e^{-tX_l} &\leqslant \left(\mathbb{E}e^{-tX_s}\right)^{(d'_L\cdot d')^{l-s}} \cdot \prod_{k=0}^{l-s-1}\left(\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\omega_{l-k}\gamma}\right)^{d'}\right)^{(d'_L\cdot d')^k}\\
&\leqslant \left[\left(\mathbb{E}e^{-tX_{s-1}}\right)^{(d'_L\cdot d')}\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\omega_s\gamma}\right)^{d'}\right]^{(d'_L\cdot d')^{l-s}} \cdot \prod_{k=0}^{l-s-1}\left(\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\omega_{l-k}\gamma}\right)^{d'}\right)^{(d'_L\cdot d')^k}\\
&= \left(\mathbb{E}e^{-tX_{s-1}}\right)^{(d'_L\cdot d')^{l-s+1}} \cdot \prod_{k=0}^{l-s}\left(\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\omega_{l-k}\gamma}\right)^{d'}\right)^{(d'_L\cdot d')^k},
\end{aligned}$$

which concludes the correctness of the induction step for a difference of $l - s + 1$. □

Based on these bounds, in the following subsection we present concrete bounds on $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$ for the BSC. This technique may be applied to other memoryless binary-input output-symmetric channels as well, e.g., an analysis for BI-AWGN channel as a generalization of the analysis presented in [HE11].

5.2 Analysis for Binary Symmetric Channel

Consider the binary symmetric channel with crossover probability p denoted by BSC(p). In the case that the all-zero codeword is transmitted, the channel input is $c_i = 0$ for every i. Hence, $\Pr\big(\lambda_i = -log\big(\frac{1-p}{p}\big)\big) = p$, and $\Pr\big(\lambda_i = +log\big(\frac{1-p}{p}\big)\big) = 1-p$. Since $\Pi_{\lambda,d,d_L,d_R}(h,\omega)$ is invariant under positive scaling of the vector λ, we consider in the following analysis the scaled vector λ in which $\lambda_i = +1$ w.p. p, and -1 w.p. $(1-p)$.

Following the ideas in the analysis of Arora *et al.* [ADS09], we apply a simple analysis in the case of uniform weight vector ω. Then, we present improved bounds by using a non-uniform weight vector.

5.2.1 Uniform Weights

Consider the case where $\omega = 1^h$. Let $c_1 \triangleq \mathbb{E}e^{-tX_0}$ and $c_2 \triangleq \binom{d'_R}{d'}(\mathbb{E}e^{-t\lambda_i})^{d'}$, and define $c \triangleq \min_{t\geqslant 0} c_1 \cdot c_2^{1/(d'_L\cdot d'-1)}$. Note that $c_1 \leqslant c_2$ (see Equation (22)).We consider the case where $c < 1$. By substituting notations of c_1 and c_2 in Lemma 16 for $s = 0$, we have

$$\begin{aligned}
\mathbb{E}e^{-tX_l} &\leqslant \left(\mathbb{E}e^{-tX_0}\right)^{(d'_L\cdot d')^l} \cdot \prod_{k=0}^{l-1}\left(\binom{d'_R}{d'}\left(\mathbb{E}e^{-t\lambda_i}\right)^{d'}\right)^{(d'_L\cdot d')^k}\\
&= {c_1}^{(d'_L\cdot d')^l} \cdot \prod_{k=0}^{l-1} {c_2}^{(d'_L\cdot d')^k}\\
&= {c_1}^{(d'_L\cdot d')^l} \cdot {c_2}^{\sum_{k=0}^{l-1}(d'_L\cdot d')^k}\\
&= {c_1}^{(d'_L\cdot d')^l} \cdot {c_2}^{\frac{(d'_L\cdot d')^l-1}{d'_L\cdot d'-1}}\\
&= \left(c_1 \cdot {c_2}^{\frac{1}{d'_L\cdot d'-1}}\right)^{(d'_L\cdot d')^l} \cdot {c_2}^{-\frac{1}{d'_L\cdot d'-1}}\\
&\leqslant c^{(d'_L\cdot d')^l-1}.
\end{aligned}$$

By Lemma 15, we conclude that

$$\Pi_{\lambda,d,d_L,d_R}(h,1^h) \leqslant c^{d_L\cdot(d'_L\cdot d')^{h-1}-d_L}.$$

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

$$X_0 = \begin{cases} d'-2k & \text{w.p. } \binom{d'_R}{k}p^k(1-p)^{d'_R-k},\ \forall k.\ 0 \leqslant k < d', \\ -d' & \text{w.p. } \sum_{k=d'}^{d'_R}\binom{d'_R}{k}p^k(1-p)^{d'_R-k}. \end{cases} \tag{24}$$

Therefore,

$$\begin{aligned} c_1(p,d,d_L,d_R,t) &= \sum_{k=0}^{d'-1}\binom{d'_R}{k}p^k(1-p)^{(d'_R-k)}e^{-t(d'-2k)} && (25)\\ &\quad+\left(\sum_{k=d'}^{d'_R}\binom{d'_R}{k}p^k(1-p)^{d'_R-k}\right)e^{td'}, \text{ and} && (26)\\ c_2(p,d,d_L,d_R,t) &= \binom{d'_R}{d'}\left((1-p)e^{-t}+pe^{t}\right)^{d'}. && (27)\end{aligned}$$

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

Lemma 17. *Let $p \in (0,\frac{1}{2})$ and let $d, d_L, d_R \geqslant 2$ s.t. $d'_L \cdot d' \geqslant 2$. Denote by c_1 and c_2 the functions defined in (25)-(27). If the following condition is satisfied*

$$c = \min_{t\geqslant 0}\left\{\left(c_1(p,d,d_L,d_R,t)\right)\cdot\left(c_2(p,d,d_L,d_R,t)\right)^{1/(d'_L\cdot d'-1)}\right\} < 1,$$

then for $h \in \mathbb{N}$ and $\omega = 1^h$, we have

$$\Pi_{\lambda,d,d_L,d_R}(h,\omega) \leqslant c^{d_L\cdot {d'_L}^{h-1}-d_L}.$$

Note that $\Pi_{\lambda,d,d_L,d_R}(h,1^h)$ decreases doubly-exponentially as a function of h.

For $(2,16)$-regular graphs and $d \in \{3,4\}$, we obtain the following corollary.

Corollary 18. *Let $d_L = 2$, and $d_R = 16$.*

1. *Let $d = 3$ and $p \leqslant 0.0067$. Then, there exists a constant $c < 1$ such that for every $h \in \mathbb{N}$ and $w = 1^h$,*
$$\Pi_{\lambda,d,d_L,d_R}(h,1^h) \leqslant c^{2^{h-1}}.$$
2. *Let $d = 4$ and $p \leqslant 0.0165$. Then, there exists a constant $c < 1$ such that for every $h \in \mathbb{N}$ and $w = 1^h$,*
$$\Pi_{\lambda,d,d_L,d_R}(h,1^h) \leqslant c^{3^{h-1}}.$$

The bound on p for which Corollary 18 applies grows with d. This fact confirms that analysis based on bigger trees, i.e., d-trees with $d > 2$ instead of skinny trees, implies better bounds on the error probability and higher lower bounds on the threshold. Also, for $d > 2$, we may apply the analysis to $(2, d_R)$-regular codes; a case that is not applicable by the analysis of Arora *et al.* [ADS09].

5.2.2 Improved Bounds Using Non-Uniform Weights

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

Lemma 19. *Let $p \in (0, \frac{1}{2})$ and let $d, d_L, d_R \geqslant 2$ s.t. $d'_L \cdot d' \geqslant 2$. Suppose that for some $s \in \mathbb{N}$ and some weight vector $\overline{\omega} \in \mathbb{R}^s_+$,*

$$\min_{t \geqslant 0} \left\{ \mathbb{E} e^{-tX_s} \right\} < \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{-\frac{1}{d'_L \cdot d' - 1}}. \tag{28}$$

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

$$\Pi_{\lambda,d,d_L,d_R}(h, \omega^{(\rho)}) \leqslant \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{-\frac{d'_L}{d'_L \cdot d' - 1}} \cdot c^{d_L \cdot (d'_L \cdot d')^{h-s-1}}.$$

Proof. By Lemma 16, we have

$$\mathbb{E} e^{-tX_{h-1}} \leqslant \left(\mathbb{E} e^{-tX_s} \right)^{(d'_L \cdot d')^{h-s-1}} \cdot \left(\binom{d'_R}{d'} \left(\mathbb{E} e^{-t\rho\eta} \right)^{d'} \right)^{\frac{(d'_L \cdot d')^{h-s-1} - 1}{d'_L \cdot d' - 1}}.$$

Note that $\mathbb{E} e^{-t\rho\eta}$ is minimized for $e^{t\rho} = \sqrt{p(1-p)}$. Hence,

$$\begin{aligned} \mathbb{E} e^{-tX_{h-1}} &\leqslant \left(\mathbb{E} e^{-tX_s} \right)^{(d'_L \cdot d')^{h-s-1}} \cdot \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{\frac{(d'_L \cdot d')^{h-s-1} - 1}{d'_L \cdot d' - 1}} \\ &\leqslant \left[\left(\mathbb{E} e^{-tX_s} \right) \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{\frac{1}{d'_L \cdot d' - 1}} \right]^{(d'_L \cdot d')^{h-s-1}} \cdot \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{-\frac{1}{d'_L \cdot d' - 1}}. \end{aligned}$$

Let $c \triangleq \min_{t \geqslant 0} \left\{ \mathbb{E} e^{-tX_s} \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{\frac{1}{d'_L \cdot d' - 1}} \right\}$. By (28), $c < 1$. Let $t^* = \arg\min_{t \geqslant 0} \mathbb{E} e^{-tX_s}$, then

$$\mathbb{E} e^{-t^* X_{h-1}} \leqslant c^{(d'_L \cdot d' - 1)^{h-s-1}} \cdot \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{-\frac{1}{d'_L \cdot d' - 1}}.$$

Using Lemma 15, we conclude that

$$\Pi_{\lambda,d,d_L,d_R}(h, \omega^{(\rho)}) \leqslant c^{d_L (d'_L \cdot d' - 1)^{h-s-1}} \cdot \left(\binom{d'_R}{d'} \left(2\sqrt{p(1-p)}\right)^{d'} \right)^{-\frac{d_L}{d'_L \cdot d' - 1}}.$$

and the lemma follows. □

Consider a weight vector $\overline{\omega}$ with components $\overline{\omega}_l = ((d_L - 1)(d - 1))^l$. This weight vector has the effect that if λ assigns the same value to every variable node, then every level in a skinny tree τ contributes equally to $val_{\overline{\omega}}(\tau; \lambda)$. For $h > s$, consider a weight vector $\omega^{(\rho)} \in \mathbb{R}^h_+$ defined by

$$\omega_l = \begin{cases} \overline{\omega}_l & \text{if } 0 \leqslant l < s, \\ \rho & \text{if } s \leqslant l < h. \end{cases}$$

s	p_0	s	p_0
0	0.0086	4	0.0164
1	0.011	5	0.0171
2	0.0139	6	0.0177
3	0.0154	10	0.0192

Table 2: Computed values of p_0 for finite s in Corollary 20. Values are presented for $(d_L, d_R) = (2, 16)$ and $d = 3$.

s	p_0	s	p_0
0	0.0218	4	0.039
1	0.0305	5	0.0405
2	0.0351	6	0.0415
3	0.0375	10	0.044

Table 3: Computed values of p_0 for finite s in Corollary 21. Values are presented for $(d_L, d_R) = (2, 16)$ and $d = 4$.

Note that the first s components of $\omega^{(\rho)}$ are non-uniform while the other components are uniform.

For a given p, d, d_L, and d_R, and for a concrete value s we can compute the distribution of X_s using the recursion in (17)-(19). Moreover, we can also compute the value $\min_{t \geqslant 0} \mathbb{E}e^{-tX_s}$. For $(2, 16)$-regular graphs and we obtain the following corollaries. Corollary 20 is stated for the case where $d = 3$, and Corollary 21 is stated for the case where $d = 4$.

Corollary 20. *Let $p \leqslant p_0$, $d = 3$, $d_L = 2$, and $d_R = 16$. For the following values of p_0 and s in Table 2 it holds that there exists a constant $c < 1$ such that for every $h > s$,*

$$\Pi_{\lambda,d,d_L,d_R}(h, \omega) \leqslant \frac{1}{420}\big(p(1-p)\big)^{-1} \cdot c^{2^{h-s}}.$$

Corollary 21. *Let $p \leqslant p_0$, $d = 4$, $d_L = 2$, and $d_R = 16$. For the following values of p_0 and s in Table 3 it holds that there exists a constant $c < 1$ such that for every $h > s$,*

$$\Pi_{\lambda,d,d_L,d_R}(h, \omega) \leqslant \frac{1}{60}\big(p(1-p)\big)^{-\frac{3}{4}} \cdot c^{3^{h-s}}.$$

Note that for a fixed s, the probability $\Pi_{\lambda,d,d_L,d_R}(h, \omega)$ decreases doubly-exponentially as a function of h. Since it's required that $s < h$, Corollaries 20 and 21 apply only to codes whose Tanner graphs have girth larger than $4h$.

5.3 Analysis for MBIOS channels

Theorem 11 generalizes to MBIOS channels as follows.

Theorem 22. *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 a Tanner code based on G with minimum local distance d^*. Consider an MBIOS channel, and suppose that $y \in \mathbb{R}^N$ is the word obtained from the channel given $c = 0^N$. Let $\lambda \in \mathbb{R}$ denote the log-likelihood ratio of the received channel observations. Then, for any (d_L, d_R) and $2 \leqslant d \leqslant d^*$ s.t. $(d_L - 1)(d-1) \geqslant 2$, LP-decoding succeeds with probability at least $1 - \exp(-N^\gamma)$ for some constant $0 < \gamma < 1$, provided that*

$$\min_{t \geqslant 0} \left\{ \mathbb{E}e^{-tX_0} \cdot \left(\binom{d_R - 1}{d-1} \left(\mathbb{E}e^{-t\lambda} \right)^{(d-1)} \right)^{\frac{1}{(d_L-1)(d-1)-1}} \right\} < 1.$$

where $X_0 = \sum_{i=1}^{d-1} \min^{[i]}\{\lambda^{(1)}, \ldots, \lambda^{(d_R-1)}\}$ where the random variables $\lambda^{(i)}$ are distributed identically and independently to λ.

6 Message-Passing Decoding with ML Guarantee for Irregular LDPC Codes

In this section we present a weighted min-sum decoder (called, NWMS) for irregular LDPC codes over memoryless binary-input output-symmetric channels. In Section 7 we prove that the decoder computes the maximum-likelihood (ML) codeword if a locally-optimal codeword exists (Theorem 23). Moreover, an ML-certificate can be computed efficiently for the output of the decoder. Note that Algorithm NWMS is not presented as a min-sum algorithm. However, in Section 7, an equivalent min-sum version is presented.

From this point on, we deal with Tanner codes based on Tanner graphs $G = \{\mathcal{V} \cup \mathcal{J}, E\}$ with parity-check local-codes. Local-code nodes $C \in \mathcal{J}$ in this case are called *check nodes*. The graph G may be either regular or irregular. Theorem 23 holds for every Tanner graph, regardless of its girth, degrees, or density.

Previous work. A huge number of works deal with message-passing decoding. We point out three works that can be viewed as precursors to our decoding algorithm. Gallager [Gal63] presented the sum-product iterative decoding algorithm for LDPC codes. Tanner [Tan81] viewed iterative decoding algorithms as message passing algorithms over the edges of the Tanner graph. Wiberg [Wib96] characterized decoding failures of the min-sum iterative decoding algorithm by negative cost trees. Message-passing decoding algorithms proceed by iterations of “ping-pong” messages between the variables nodes and the local-code nodes in the Tanner graph. These messages are sent only along the edges.

Algorithm description. Algorithm NWMS(λ, h, w), listed as Algorithm 2, is a normalized w-weighted version of the min-sum algorithm for decoding Tanner codes with parity-check local-codes. The input to algorithm NWMS consists of an LLR vector $\lambda \in \mathbb{R}^N$, an integer $h > 0$ that determines the number of iterations, and a nonnegative weight vector $w \in \mathbb{R}_+^h$. For each edge (v, C), each iteration consists of one message from the variable node v to the check node C (that is, the “ping” message), and one message from C to v (that is, the “pong” message). Hence, the time and message complexity of Algorithm 2 is $O(|E| \cdot h)$.

Let $\mu_{v \to C}^{(l)}$ denote the “ping” message from a variable node $v \in \mathcal{V}$ to an adjacent check-node $C \in \mathcal{J}$ in iteration l of the algorithm. Similarly, let $\mu_{C \to v}^{(l)}$ denotes the “pong” message from

$C \in \mathcal{J}$ to $v \in \mathcal{V}$ in iteration l. Denote by μ_v the final value computed by variable node $v \in \mathcal{V}$. The output of the algorithm $\hat{x} \in \{0,1\}^N$ is computed locally by each variable node in Line 12.

Algorithm NWMS may be applied to any memoryless binary-input output-symmetric channel (e.g., BEC, BSC, AWGN, etc.) because the input is the LLR vector.

Algorithm 2 NWMS(λ, h, w) - An iterative normalized weighted min-sum decoding algorithm. Given an LLR vector $\lambda \in \mathbb{R}^N$ and level weights $w \in \mathbb{R}_+^h$, outputs a binary string $\hat{x} \in \{0,1\}^N$.

```
 1: Initialize: $\forall C \in \mathcal{J}$, $\forall v \in \mathcal{N}(C) : \mu_{C\to v}^{(-1)} \leftarrow 0$
 2: for $l = 0$ to $h-1$ do
 3:     for all $v \in \mathcal{V}$, $C \in \mathcal{N}(v)$ do {"PING"}
 4:         $\mu_{v\to C}^{(l)} \leftarrow \frac{w_{h-l}}{\deg_G(v)}\lambda_v + \frac{1}{\deg_G(v)-1}\sum_{C' \in \mathcal{N}(v)\setminus\{C\}} \mu_{C'\to v}^{(l-1)}$
 5:     end for
 6:     for all $C \in \mathcal{J}$, $v \in \mathcal{N}(C)$ do {"PONG"}
 7:         $\mu_{C\to v}^{(l)} \leftarrow \left(\prod_{v' \in \mathcal{N}(C)\setminus\{v\}} \mathrm{sign}(\mu_{v'\to C}^{(l)})\right) \cdot \min_{v' \in \mathcal{N}(C)\setminus\{v\}} \{|\mu_{v'\to C}^{(l)}|\}$
 8:     end for
 9: end for
10: for all $v \in \mathcal{V}$ do {Decision}
11:     $\mu_v \leftarrow w_0\lambda_v + \sum_{C \in \mathcal{N}(v)} \mu_{C\to v}^{(h-1)}$
12:     $\hat{x}_v \leftarrow \begin{cases} 0 & \text{if } \mu_v^{(h-1)} > 0, \\ 1 & \text{otherwise.} \end{cases}$
13: end for
```

The following theorem states that NWMS(λ, h, w) computes an $(h, w, 2)-$locally-optimal codeword for λ if such a codeword exists. The theorem implies that there exists at most one $(h, w, 2)$-locally optimal codeword. The proof of the theorem appears in Section 7.

Theorem 23 (NWMS guaranties local-optimality)**.** *Let $G = (\mathcal{V} \cup \mathcal{J}, E)$ denote a Tanner graph and let $\mathcal{C}(G) \subset \{0,1\}^N$ denote the corresponding Tanner code with parity-check local-codes. Let $h \in \mathbb{N}_+$ and let $w \in \mathbb{R}_+^h$ denote a non-negative weight vector. Let $\lambda \in \mathbb{R}^N$ denote the LLR vector of the channel output. If $x \in \mathcal{C}(G)$ is an $(h, w, 2)$-locally optimal codeword for λ, then the output $\hat{x}$ of* NWMS(λ, h, w) *equals x.*

The dynamic programming algorithm described in Section 3.1 can be used to verify whether NWMS(λ, h, w) outputs an $(h, w, 2)$-locally optimal codeword. If so, then, by Theorem 5, the output of NWMS(λ, h, w) is the unique ML-codeword.

Corollary 32 states that for MBIOS channels, the probability that NWMS fails is independent of the transmitted codeword. Hence, the following corollary is a contra-positive of Theorem 23 provided the all-zero codeword assumption.

Corollary 24. *Assume that the all-zero codeword was transmitted, and let $\lambda \in \mathbb{R}^N$ denote the log-likelihood ratio for the received word. If NWMS(λ, h, w) fails to decode the all-zero codeword for $w \in \mathbb{R}_+^h \backslash \{0^h\}$, then there exists a vector $\beta \in \mathcal{B}_2^{(w)}$ such that $\langle \lambda, \beta \rangle \leqslant 0$.*

We therefore have for a fixed h and $w \in \mathbb{R}_+^h \backslash \{0^h\}$ that

$$\Pr\{\mathrm{NWMS}(\lambda, h, w) \text{ fails}\} \leqslant \Pr\left\{\exists \beta \in \mathcal{B}_2^{(w)} \text{ such that } \langle \lambda, \beta \rangle \leqslant 0 \middle| c = 0^N\right\}. \tag{29}$$

Following Equation (29), we note that for the case of regular LDPC codes, the previous bounds on the probability that a local-optimality certificate exists [ADS09, HE11] also apply to the probability of NWMS decoding success. For example, consider $(3,6)$-regular LDPC codes whose Tanner graphs G have logarithmic girth, let $h = \frac{1}{4} girth(G)$ and define a constant weight vector $w = 1^h$. Then, $\text{NWMS}(\lambda, h, w)$ succeeds in recovering the transmitted codeword with probability at least $1 - \exp(-n^\gamma)$ for some constant $0 < \gamma < 1$ in the following cases: (1) In a BSC with crossover probability $p < 0.05$ (implied by [ADS09, Theorem 5]). (2) In a BI-AWGN channel with $\frac{E_b}{N_0} \geqslant 2.67$dB (implied by [HE11, Theorem 1]).

It remains to explore good weighting schemes (choice of vectors w) and prove bounds on the success probability of the NWMS decoder for specific families of irregular LDPC codes.

7 Proof of Theorem 23

Proof outline. The proof of Theorem 23 is based on two observations. (1) We present an equivalent algorithm, called NWMS2 (Section 7.1). It is easier to prove that Algorithm NWMS2 outputs the all-zero codeword if 0^N is locally optimal (Sections 7.2-7.3). (2) In Lemma 31 we prove that algorithm NWMS is symmetric (Section 7.4). The symmetry characterization provides a mapping from every pair (x, λ) of a codeword and an LLR vector to a pair $(0^N, \lambda^0)$ of the all-zero codeword and a corresponding LLR vector λ^0.

The proof of Theorem 23 is obtained as follows. We prove the contrapositive statement, that is, if $x \neq \text{NWMS}(\lambda, h, w)$, then x is not $(h, w, 2)$-locally optimal for λ. Let x denote a codeword, and define $b \in \{\pm 1\}^N$ by $b_i \triangleq (-1)^{x_i}$. Let "$*$" denote a coordinate-wise vector multiplication. Define $\lambda^0 \triangleq b * \lambda$, so $\lambda = b * \lambda^0$. The proof is obtained by the following derivations:

$$
\begin{array}{ll}
 x \neq \text{NWMS}(\lambda, h, w) & \\
\Rightarrow x \neq \text{NWMS}(b * \lambda^0, h, w) & \\
\Rightarrow x \neq x \oplus \text{NWMS}(\lambda^0, h, w) & [\text{Lemma } 31,\ \text{symmetry}] \\
\Rightarrow 0^N \neq \text{NWMS}(\lambda^0, h, w) & \\
\Rightarrow \exists \beta \in \mathcal{B}_2^{(w)}. \langle \lambda^0, \beta \rangle \leqslant 0 & [\text{Lemma } 28,\ \text{local optimality (LO)}] \\
\Rightarrow \langle b * \lambda^0, x \oplus \beta \rangle \leqslant \langle b * \lambda^0, x \rangle & [\text{Lemma } 29,\ \text{mapping preserves LO}] \\
\Rightarrow \langle \lambda, x \oplus \beta \rangle \leqslant \langle \lambda, x \rangle & \\
\Rightarrow x \text{ is not } (h, w, 2)\text{−locally optimal for } \lambda. & \text{QED}
\end{array}
$$

We now prove the three lemmas used in the foregoing proof.

7.1 NWMS2 : An Equivalent Version

The normalized weighted min-sum decoding algorithm presented in section 6 is input the log-likelihood ratio. We refer to this algorithm as a min-sum algorithm in light of the general description of Wiberg [Wib96]. In Wiberg's description, every check node finds a minimum value from a set of functions on the incoming messages, and every variable node computes the sum of the incoming messages and its corresponding channel observation. Hence the name "min-sum".

Let $y \in \mathbb{R}^N$ denote channel observations. For $a \in \{0,1\}$, define the log-likelihood of y_i by $\lambda_i(a) \triangleq -\log\big(\Pr(y_i|c_i = a)\big)$. Note that the log-likelihood ratio λ_i for y_i equals $\lambda_i(1) - \lambda_i(0)$.

Algorithm $\textsc{nwms2}(\lambda(0), \lambda(1), h, w)$, listed as Algorithm 3, is a normalized w-weighted min-sum algorithm. Algorithm $\textsc{nwms2}$ computes separate "reliabilities" for "0" and "1". Namely, $\mu^{(l)}_{v\to C}(a)$ and $\mu^{(l)}_{C\to v}(a)$ denote the messages corresponding to the assumption that node v is assigned the value a (for $a \in \{0,1\}$).

Line 7 takes the main difference between the presentations of Algorithm 2 and Algorithm 3. The computation in Line 7 of the message $\mu^{(l)}_{C\to v}(a)$ from check node C to variable node v proceeds as follows. Consider assignments $x \in \{0,1\}^{\deg(C)}$ to variable nodes adjacent to C with even weight, i.e., parity local codewords, such that $x_v = a$. For every such assignment x, the check node C computes the sum of the incoming messages $\mu^{(l)}_{u\to C}(x_u)$ from the neighboring nodes $u \in \mathcal{N}(C) \setminus \{v\}$ other than v, according to their assignment x_u by x. Then, the message $\mu^{(l)}_{C\to v}(a)$ equals to the minimum value over the valid summations.

Algorithm 3 NWMS2$(\lambda(0), \lambda(1), h, w)$ - An iterative normalized weighted min-sum decoding algorithm. Given an log-likelihood vectors $\lambda(a) \in \mathbb{R}^N$ for $a \in \{0,1\}$ and level weights $w \in \mathbb{R}^h_+$, outputs a binary string $\hat{x} \in \{0,1\}^N$.

```
1: Initialize: $\forall C \in \mathcal{J}, \forall v \in \mathcal{N}(C), \forall a \in \{0,1\} : \mu^{(-1)}_{C\to v}(a) \leftarrow 0$
2: for $l = 0$ to $h-1$ do
3:     for all $v \in \mathcal{V}$, $C \in \mathcal{N}(v)$, $a \in \{0,1\}$ do {"PING"}
4:         $\mu^{(l)}_{v\to C}(a) \leftarrow \frac{w_{h-l}}{\deg_G(v)}\lambda_v(a) + \frac{1}{\deg_G(v)-1}\sum_{C' \in \mathcal{N}(v)\setminus\{C\}} \mu^{(l-1)}_{C'\to v}(a)$
5:     end for
6:     for all $C \in \mathcal{J}$, $v \in \mathcal{N}(C)$, $a \in \{0,1\}$ do {"PONG"}
7:         $\mu^{(l)}_{C\to v}(a) \leftarrow \min_{\{x \in \{0,1\}^{\deg(C)} : |x| \text{ is even and } x_v = a\}} \big\{ \sum_{v' \in \mathcal{N}(C)\setminus\{v\}} \mu^{(l)}_{v'\to C}(x_{v'}) \big\}$
8:     end for
9: end for
10: for all $v \in \mathcal{V}$ do {Decision}
11:     $\mu_v(a) \leftarrow w_0\lambda_v(a) + \sum_{C \in \mathcal{N}(v)} \mu^{(h-1)}_{C\to v}(a)$
12:     $\hat{x}_v \leftarrow \begin{cases} 0 & \text{if } \big(\mu_v^{(h-1)}(1) - \mu_v^{(h-1)}(0)\big) > 0, \\ 1 & \text{otherwise.} \end{cases}$
13: end for
```

Following Wiberg [Wib96, Appendix A.3], we claim that Algorithms 2 and 3 are equivalent.

Claim 25. *Let λ, $\lambda(0)$, and $\lambda(1)$ in $\mathbb{R}^N$ denote the LLR vector and the two log-likelihood vectors for a channel output $y \in \mathbb{R}^N$. Then, for every $h \in \mathbb{N}_+$ and $w \in \mathbb{R}^h_+$, the following equalities hold:*

1. $\mu^{(l)}_{v\to C} = \mu^{(l)}_{v\to C}(1) - \mu^{(l)}_{v\to C}(0)$ *and* $\mu^{(l)}_{C\to v} = \mu^{(l)}_{C\to v}(1) - \mu^{(l)}_{C\to v}(0)$ *in every iteration l.*
2. $\mu_v = \mu_v(1) - \mu_v(0)$. *Hence* $\textsc{nwms}(\lambda, h, w)$ *and* $\textsc{nwms2}(\lambda(0), \lambda(1), h, w)$ *output the same vector $\hat{x}$.*

7.2 NWMS2 as a Dynamic Programming Algorithm

In Lemma 26 we prove that Algorithm $\textsc{nwms2}$ is a dynamic programming algorithm that computes, for every variable node v, two min-weight valid configurations. We now elaborate on the definition of valid configurations and their weight.

Fix a variable node $r \in \mathcal{V}$. We refer to r as the *root*. Trace the messages that lead to the decision $\hat{x}_r$ in NWMS2. Namely, consider the path-prefix tree rooted at r consisting of all the paths of length $2h$ ending at r. (Note that these paths may not zigzag, hence an edge (u,v) and its reversal (v,u) may not appear consecutively.) Denote this path-prefix tree by $\mathcal{T}_r^{2h}$. The variable nodes and check nodes in $\mathcal{T}_r^{2h}$ are denoted by $\hat{\mathcal{V}}$ and $\hat{\mathcal{J}}$, respectively.

Every binary vector $x \in \{0,1\}^{|\hat{\mathcal{V}}|}$ defines an assignment to variable nodes in $\hat{\mathcal{V}}$. We say that x is a *valid configuration* if it satisfies all parity-checks in $\hat{\mathcal{J}}$. Namely, for every check node $C \in \hat{\mathcal{J}}$, the assignment to its neighbors has an even number of ones.

The weight of a valid configuration x is defined using the following functions. (1) Extend the log-likelihood functions to the variable nodes in $\hat{\mathcal{V}}$ by $\lambda_{\hat{v}} \triangleq \lambda_v$, where $\hat{v} \sim v$. (2) Assign weights to levels of variable nodes by a vector $w = (w_1, \ldots, w_h) \in \mathbb{R}_+^h$. (3) Define the weight of a node $\hat{v}$ in $\mathcal{T}_r^{2h}$ with respect to w by

$$\mathcal{W}_r(\hat{v}) \triangleq \frac{w_t}{\deg_G(v)} \cdot \prod_{\hat{u} \in (P_{r\hat{v}} \cap \hat{\mathcal{V}}) \setminus \{r, \hat{v}\}} \frac{1}{\deg_G(u) - 1},$$

where $t = \frac{d(r,\hat{v})}{2}$, $\hat{u} \sim u$, and $\hat{v} \sim v$.

The weight of a valid configuration x is defined by

$$\mathcal{W}_r(x) \triangleq \sum_{\hat{v} \in \hat{\mathcal{V}}} \lambda_{\hat{v}}(x_{\hat{v}}) \cdot \mathcal{W}_r(\hat{v}).$$

The following lemma characterizes NWMS2 as a computation of min-weight configurations.

Lemma 26. *Let $\hat{x}$ denote the output of* NWMS2$(\lambda(0), \lambda(1), h, w)$. *Let $z(v)$ denote a valid configuration in $\mathcal{T}_v^{2h}$ with minimum $\mathcal{W}_v$ weight. Then, $\hat{x}_v = \big(z(v)\big)_v$.*

Proof sketch. The proof of Lemma 26 is obtained by induction on the number of iterations. The key idea is that a message $\mu_{v \to C}^{(l)}(a)$ [Line 4] at iteration l equals to the minimum weight of a valid subconfiguration on the subtree of hight $2l$ hanging from v, that assigns v the value a. The computation of message $\mu_{C \to v}^{(l)}(a)$ in Line 7 plays the main rule in the proof of the inductive step. Under the assumption that v is assigned the value a, for every local valid assignment to its neighbors, check node C accumulates the messages received from its children that correspond to the local valid assignment. By the induction hypothesis, the values of the messages received from the children of C equal the min-weight valid subconfiguration hanging from them. By choosing the minimum valid summation, $\mu_{C \to v}^{(l)}(a)$ equals the minimum weight of a valid subconfiguration hanging from C that assigns v the value a. □

Define the $\mathcal{W}^*$ cost of a configuration x in $\mathcal{T}_r^{2h}$ by

$$\mathcal{W}_r^*(x) \triangleq \sum_{\hat{v} \in \hat{\mathcal{V}}} \lambda_{\hat{v}} \cdot \mathcal{W}_r(\hat{v}) \cdot x_{\hat{v}}.$$

Note that $\mathcal{W}_r^*(x)$ uses the LLR vector λ (i.e., $\lambda_{\hat{v}} = \lambda_{\hat{v}}(1) - \lambda_{\hat{v}}(0)$).

Corollary 27. *Let $\hat{x}$ denote the output of* NWMS(λ, h, w). *Let $z^*(v)$ denote a valid configuration in $\mathcal{T}_v^{2h}$ with minimum $\mathcal{W}^*$ cost. Then, $\hat{x}_v = \big(z^*(v)\big)_v$.*

Proof. Let $\hat{x} = \text{NWMS}(\lambda, h, w)$ and $\hat{z} = \text{NWMS}2(\lambda(0), \lambda(1), h, w)$. By Claim 25, $\hat{x}_v = \hat{z}_v$ for every $v \in \mathcal{V}$. Therefore,

$$\begin{aligned}
\hat{x}_v = \hat{z}_v &= \operatorname*{arg\,min}_{\text{valid } x \in \mathcal{T}_v^{2h}} \mathcal{W}_v(x) \\
&= \operatorname*{arg\,min}_{\text{valid } x \in \mathcal{T}_v^{2h}} \left\{ \mathcal{W}_v(x) - \mathcal{W}_v(0^{|\hat{\mathcal{V}}|}) \right\} \\
&= \operatorname*{arg\,min}_{\text{valid } x \in \mathcal{T}_v^{2h}} \left\{ \sum_{\hat{u} \in \hat{\mathcal{V}}: x_{\hat{u}}=1} \lambda_{\hat{u}}(1) \cdot \mathcal{W}_v(\hat{u}) - \sum_{\hat{u} \in \hat{\mathcal{V}}: x_{\hat{u}}=1} \lambda_{\hat{u}}(0) \cdot \mathcal{W}_v(\hat{u}) \right\} \\
&= \operatorname*{arg\,min}_{\text{valid } x \in \mathcal{T}_v^{2h}} \sum_{\hat{u} \in \hat{\mathcal{V}}} \lambda_{\hat{u}} \cdot \mathcal{W}_v(\hat{u}) \cdot x_{\hat{u}} = \operatorname*{arg\,min}_{\text{valid } x \in \mathcal{T}_v^{2h}} \mathcal{W}_v^*(x).
\end{aligned}$$

The second line relies on the fact that $\mathcal{W}_v(0^{|\hat{\mathcal{V}}|})$ is a constant. The elements $\lambda_{\hat{u}}(\hat{x}_{\hat{u}}) \cdot \mathcal{W}_v(\hat{u})$ in $\mathcal{W}_v(x)$ where $x_{\hat{u}} = 0$ are reduced by the substraction of the same elements in $\mathcal{W}_v(0^{|\hat{\mathcal{V}}|})$, leaving in the third line only elements that correspond to bits $x_{\hat{u}} = 1$. The fourth line is obtained by the LLR definition $\lambda_{\hat{u}} = \lambda_{\hat{u}}(1) - \lambda_{\hat{u}}(0)$. □

7.3 Connections to Local Optimality

For two vectors $x, y \in \mathbb{R}^k$, let "$*$" denote coordinatewise multiplication, i.e., $x * y \triangleq (x_1 \cdot y_1, \ldots, x_k \cdot y_k)$.

The following lemma implies that NWMS algorithm outputs the all-zero codeword if 0^N is locally optimal.

Lemma 28. *Let $\hat{x}$ denote the output of* $\text{NWMS}(\lambda, h, w)$*. If $\hat{x}_v = 1$, then there exists a deviation $\beta \in \mathcal{B}_2^{(w)}$ corresponding to a w-weighted 2-tree such that $\langle \lambda, \beta \rangle \leqslant 0$.*

Proof. Assume that $\hat{x}_v = 1$, and consider $\mathcal{T}_v^{2h} = (\hat{\mathcal{V}} \cup \hat{\mathcal{J}}, \hat{E})$. Then, by Corollary 27, there exists a valid configuration $z^* \in \{0,1\}^{|\hat{\mathcal{V}}|}$ in $\mathcal{T}_v^{2h}$ with $z_v^* = 1$ that satisfies

$$\forall \text{valid configuration } u \in \mathcal{T}_v^{2h}.\ \mathcal{W}_v^*(z^*) \leqslant \mathcal{W}_v^*(u). \tag{30}$$

Let $\mathcal{T}(z^*)$ denote the subgraph of $\mathcal{T}_v^{2h}$ induced by $\hat{\mathcal{V}}(z^*) \cup \mathcal{N}\big(\hat{\mathcal{V}}(z^*)\big)$ where $\hat{\mathcal{V}}(z^*) = \{\hat{u} \in \hat{\mathcal{V}} | z_{\hat{u}}^* = 1\}$. Note that $\mathcal{T}(z^*)$ is a forest. Because $z_v^* = 1$ and z^* is a valid configuration, the forest $\mathcal{T}(z^*)$ must contain a 2-tree of height $2h$ rooted at the node v; denote this tree by $\mathcal{T}[v, 2h, 2]$. Let $\tau \in \{0,1\}^{|\hat{\mathcal{V}}|}$ denote the support of $\mathcal{T}[v, 2h, 2]$, and let $z^0 \in \{0,1\}^{|\hat{\mathcal{V}}|}$ denote the support of $\mathcal{T}(z^*) \setminus \mathcal{T}[v, 2h, 2]$. Then, $z^* = \tau + z^0$, where z^0 is also necessarily a valid configuration. By linearity, we have

$$\mathcal{W}_v^*(z^*) = \mathcal{W}_v^*(\tau + z^0) = \mathcal{W}_v^*(\tau) + \mathcal{W}_v^*(z^0). \tag{31}$$

Because z^0 is a valid configuration, by Equation (30), we have $\mathcal{W}_v^*(z^*) \leqslant \mathcal{W}_v^*(z^0)$. By Equation (31), $\mathcal{W}_v^*(\tau) \leqslant 0$.

Let $\mathcal{W}_v(\hat{\mathcal{V}}) * \tau \in R^{|\hat{\mathcal{V}}|}$ denote the vector whose component indexed by $\hat{u} \in \hat{\mathcal{V}}$ equals $\mathcal{W}_v(\hat{u}) \cdot \tau_{\hat{u}}$. The vector $\mathcal{W}_v(\hat{\mathcal{V}}) * \tau$ represents the w-weighted 2-tree $\mathcal{T}^{(w)}[v, 2h, 2]$ according to Definition 3. Hence, $\beta = \pi_G[\mathcal{T}^{(w)}[v, 2h, 2]] \in \mathcal{B}_2^{(w)}$ satisfies $\langle \lambda, \beta \rangle = \mathcal{W}_v^*(\tau) \leqslant 0$. □

The following lemma implies that x is locally optimal with respect to λ iff 0^N is locally optimal with respect to $b{*}\lambda$, where $b_i = (-1)^{x_i}$. Hence we refer to mapping $(x, \lambda) \mapsto (0^N, b{*}\lambda)$ as a mapping that preserves local optimality.

Lemma 29. *Let $x \in \{0,1\}^N$ and define $b \in \{\pm 1\}^N$ by $b_i = (-1)^{x_i}$. Then,*

$$\forall \lambda \in \mathbb{R}^N.\ \forall \beta \in [0,1]^N. \quad \langle \lambda, \beta \rangle = \langle b * \lambda, x \oplus \beta \rangle - \langle b * \lambda, x \rangle. \tag{32}$$

Proof. For $u \in [0,1]^N$, it holds that $\langle \lambda, x \oplus u \rangle = \langle \lambda, x \rangle + \sum_{i=1}^{N} (-1)^{x_i} \lambda_i u_i$. Then,

$$\begin{aligned} \langle b * \lambda, x \oplus \beta \rangle &= \langle b * \lambda, x \rangle + \sum_{i=1}^{N} (-1)^{x_i} b_i \lambda_i \beta_i \\ &= \langle b * \lambda, x \rangle + \sum_{i=1}^{N} (-1)^{x_i} (-1)^{x_i} \lambda_i \beta_i \\ &= \langle b * \lambda, x \rangle + \langle \lambda, \beta \rangle \end{aligned}$$

□

7.4 Symmetry and the All-Zero Codeword Assumption

We define symmetric decoding algorithms (see [RU08, Definition 4.81] for a discussion of symmetry in message passing algorithms).

Definition 30 (symmetry of decoding algorithm)**.** *Let $x \in \mathcal{C}$ denote a codeword and let $b \in \{\pm 1\}^N$ denote a vector defined by $b_i = (-1)^{x_i}$. Let λ denote an LLR vector. A decoding algorithm, $\textsc{dec}(\lambda)$, is* symmetric *with respect to code $\mathcal{C}$, if*

$$\forall x \in \mathcal{C}. \ \ x \oplus \textsc{dec}(\lambda) = \textsc{dec}(b * \lambda). \tag{33}$$

The following lemma states that NWMS algorithm is symmetric. The proof is by induction on the number of iterations.

Lemma 31 (symmetry of NWMS)**.** *Fix $h \in \mathbb{N}_+$ and $w \in \mathbb{R}_+^N$. Consider $\lambda \in \mathbb{R}^N$ and a codeword $x \in \mathcal{C}(G)$. Let $b \in \{\pm 1\}^N$ denote a vector defined by $b_i = (-1)^{x_i}$. Then,*

$$x \oplus \textsc{nwms}(\lambda, h, w) = \textsc{nwms}(b * \lambda, h, w). \tag{34}$$

The following corollary follows from Lemma 31 and the symmetry of an MBIOS channel (see also [RU08, Lemma 4.90]).

Corollary 32 (All-zero codeword assumption)**.** *Fix $h \in \mathbb{N}_+$ and $w \in \mathbb{R}_+^N$. For MBIOS channels, the probability that NWMS fails is independent of the transmitted codeword. That is,*

$$\Pr\{\textsc{nwms} \text{ decoding fails}\} = \Pr\left\{\textsc{nwms}(\lambda, h, w) \neq 0^N | c = 0^N\right\}.$$

8 Conclusions

We present a new combinatorial characterization of local-optimality for Tanner codes with respect to any MBIOS channel. This characterization provides an ML-certificate and an LP-certificate for a given codeword. Two applications of local-optimality are presented based on this new characterization. (i) Bounds for LP-decoding failure are proved in the case of regular Tanner codes. (ii) A new message passing decoding algorithm for irregular LDPC codes, called NWMS, is presented. The NWMS algorithm is guaranteed to find the locally optimal codeword if such exists.

An open problem is to prove for irregular Tanner codes that a locally optimal codeword exists with high probability provided that the noise is bounded. Such a result would imply that the efficient NWMS decoding algorithm is a good replacement for LP-decoding. It seems that this requires adjusting the weights w according to the Tanner graph.

References

[ADS09] S. Arora, C. Daskalakis, and D. Steurer, "Message passing algorithms and improved LP decoding," in *Proc. 41st Annual ACM Symp. Theory of Computing (STOC'09)*, Bethesda, MD, USA, pp. 3–12, 2009.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *in Proc. IEEE Int. Conf. on Communications (ICC'93), Geneva, Switzerland*, vol. 2, pp. 1064–1070, 1993.

[BMvT78] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384-386, May 1978.

[BZ02] A. Barg and G. Zémor, "Error exponents of expander codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1725–1729, Jun. 2002.

[BZ04] A. Barg and G. Zémor, "Error exponents of expander codes under linear-complexity decoding," *SIAM J. Discr. Math.*, vol. 17, no. 3, pp 426–445, 2004.

[CDE^{+}05] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288 – 1299, Aug. 2005.

[CF02] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-Based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208 –210, May 2002.

[Fel03] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, MIT, Cambridge, MA, 2003.

[FK00] B.J. Frey and R. Koetter, "Exact inference using the attenuated max-product algorithm," In *Advanced Mean Field Methods: Theory and Practice*, Cambridge, MA: MIT Press, 2000.

[FS05] J. Feldman and C. Stein, “LP decoding achieves capacity,” in *Proc. Symp. Discrete Algorithms (SODA’05)*, Vancouver, Canada, Jan. 2005, pp. 460–469.

[FWK05] J. Feldman, M. J. Wainwright, and D. R. Karger, “Using linear programming to decode binary linear codes,” *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954-972, Mar. 2005.

[Gal63] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.

[GB10] I. Goldenberg and D. Burshtein, “Error bounds for repeat-accumulate codes decoded via linear programming,” in *Proc. 6th Intern. Symp. on Turbo Codes and Iter. Inform. Proc. (ISTC‘10)*, Brest, France, pp. 43–47, Sep. 6–10, 2010.

[HE11] N. Halabi and G. Even, “LP decoding of regular LDPC codes in memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 887–897, Feb. 2011.

[JP10] Y.-Y. Jian and H.D. Pfister, “Convergence of weighted min-sum decoding via dynamic programming on coupled trees,” in *Proc. 6th Intern. Symp. on Turbo Codes and Iter. Inform. Proc. (ISTC‘10)*, Brest, France, pp. 487–491, Sep. 6–10, 2010.

[KV06] R. Koetter and P. O. Vontobel, “On the block error probability of LP decoding of LDPC codes,” in *Proc. Inaugural Workshop of the Center for Information Theory and its Applications*, La Jolla, CA, USA, Feb. 2006.

[LMSS01] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low-density parity-check codes using irregular graphs,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp.585 –598, Feb. 2001.

[Mac99] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp.399 –431, Mar. 1999.

[RU01] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[RU08] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, New York, NY, 2008.

[SR03] V. Skachek and R. M. Roth, “Generalized minimum distance iterative decoding of expander codes,” In *Proc. IEEE Information Theory Workshop (ITW’03)*, pp. 245 – 248, 2003.

[SS96] M. Sipser and D. A. Spielman, “Expander codes”, *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[Tan81] R. M. Tanner, “A recursive approach to low-complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[VK05] P. O. Vontobel and R. Koetter, Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes, *CoRR*, `http://www.arxiv.org/abs/cs.IT/0512078`, Dec. 2005.

[Von10] P. Vontobel, “A factor-graph-based random walk, and its relevance for LP decoding analysis and Bethe entropy characterization,” in *Proc. Information Theory and Applications Workshop*, UC San Diego, LA Jolla, CA, USA, Jan. 31-Feb. 5, 2010.

[Wib96] N. Wiberg, “Codes and decoding on general graphs”, Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.

[WLK95] N. Wiberg, H.-A. Loeliger, and R. Kötter, “Codes and iterative decoding on general graphs,” *Eur. Trans. Telecomm.*, vol. 6, no. 5, pp. 513–525, 1995.

[Ź01] G. Zémor, “On expander codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 835–837, Feb. 2001.