Message Passing Algorithms and Improved LP Decoding

Sanjeev Arora[1]
CS, Princeton Universty

and
Constantinos Daskalakis[2]
EECS and CSAIL, MIT

and
David Steurer
CS, Cornell University[3]

ABSTRACT

Linear programming decoding for low-density parity check codes (and related domains such as compressed sensing) has received increased attention over recent years because of its practical performance —coming close to that of iterative decoding algorithms— and its amenability to finite-blocklength analysis. Several works starting with the work of Feldman et al. showed how to analyze LP decoding using properties of expander graphs. This line of analysis works for only low error rates, about a couple of orders of magnitude lower than the empirically observed performance. It is possible to do better for the case of random noise, as shown by Daskalakis et al. and Koetter and Vontobel.

Building on work of Koetter and Vontobel, we obtain a novel understanding of LP decoding, which allows us to establish a 0.05-fraction of correctable errors for rate-$1/2$ codes; this comes very close to the performance of iterative decoders and is significantly higher than the best previously noted correctable bit error rate for LP decoding. Our analysis exploits an explicit connection between LP decoding and message passing algorithms and, unlike other techniques, directly works with the primal linear program.

An interesting byproduct of our method is a notion of a "locally optimal" solution that we show to always be globally optimal (i.e., it is the nearest codeword). Such a solution can in fact be found in near-linear time by a "re-weighted" version of the min-sum algorithm, obviating the need for linear programming. Our analysis implies, in particular, that this re-weighted version of the min-sum decoder corrects up to a 0.05-fraction of errors.

[1] Supported by NSF CCF-0832797, 0830673, and 0528414.

[2] Supported by NSF CCF-0953960 (CAREER), NSF CCF-1101491 and a Sloan Research Fellowship. Part of this work was done while the author was a postdoctoral researcher at Microsoft Research, New England.

[3] Part of this work was done while the author was at Princeton University, supported by NSF CCF-0832797, 0830673, and 0528414, and a postdoctoral researcher at Microsoft Research, New England.

1. INTRODUCTION

Low density parity-check (*LDPC*) codes are linear codes over $\mathbb{F}_2$ (or its extensions) whose constraint graph is sparse. They were introduced and analyzed by Gallager [1963] in a paper that was forgotten for several decades and recalled again only in the 1990s. Sipser and Spielman [1996] studied a subclass of these codes in which the constraint graph has good expansion properties. For these *expander codes*, they showed that a very simple *bit-flipping* strategy, originally suggested by Gallager, corrects efficiently an $\Omega(1)$ fraction of (worst-case) errors, though the actual constants were quite weak. (Myriad extensions of expander codes have been studied but will not be discussed here.)

Meanwhile, researchers in information theory rediscovered Gallager's ideas and began to view the decoding problem for LDPC codes as an example of *Maximum a posteriori* (MAP) estimation in factor graphs (a notion that also became popular in machine learning). Various iterative message-passing algorithms—two popular ones being *belief propagation* (BP) and the *min-sum algorithm*—were found to empirically yield excellent decoding performance. A survey of LDPC codes and decoding algorithms appears in [Shokrollahi 2004; Richardson and Urbanke 2001].

In a seminal paper, Richardson and Urbanke [2001], aided with some computer calculations, were able to establish that a BP-like algorithm can decode with high probability a $(3, 6)$-regular LDPC code on a binary symmetric channel with error rate up to 0.084. This is the best bound known for any decoding algorithm, and is not far from the empirically observed performance of BP, and the information theoretic limit of roughly 0.11.

Our paper concerns the *linear programming (LP) decoding method.* This method was introduced by Feldman, Wainwright and Karger [2005], in a paper that only establishes a sub-linear number of correctable errors, but also notes that the empirical performance of LP decoding is similar to that of message-passing algorithms. A subsequent paper of Feldman et al. [2007] showed that the method corrects $\Omega(1)$-fraction of (worst-case) errors for expander codes. The proof consists of constructing a dual solution, inspired by Sipser and Spielman's analysis, and yields similar bounds: a tolerance of adversarial error rate up to 0.00017. Note that the advantage of LP decoding over message-passing decoders is that in the case of decoding success, the linear program provides a certificate that the output is indeed the nearest codeword.

Unfortunately, it has remained difficult to improve the analysis of LP decoding to establish bounds closer to the empirically observed performance. Daskalakis et al. [2008] were able to show a tolerance to error rate up to 0.002 —an order of magnitude better than the bounds of Feldman et al. [2007] but still more than 40 times lower than the Richardson–Urbanke [2001] bound of 0.084 for belief propagation. Their proof constructs a more intricate dual LP solution than Feldman et al.'s, but it is still based on expansion arguments. (Note: All the bounds in this paper are quoted for (regular) rate 1/2 codes and the Binary Symmetric Channel.)

Intuitively, the main reason for the small bit error rates in the above analyses of LP decoding was that these analyses were close in spirit to the Sipser and Spielman expansion-based approach. By contrast the Richardson–Urbanke style analysis of message passing algorithms relies upon the *high girth* of the graph defining the code (specifically, the fact that high-girth graphs look locally like trees).

Nevertheless, it remained unclear how to bring girth-based arguments into the context of LP decoding. In a recent paper, Koetter and Vontobel [2006] achieved this. Their key idea was to use the min-sum algorithm rather than Belief-Propagation (which uses highly nonlinear operations). They showed how to transform the messages exchanged in the min-sum algorithm into an intricate dual solution. (Their construction was inspired by the Gauss–Seidel method to solve convex programs.) Though they did not report any numbers in their paper, our calculations show that their analysis of LP decoding allows $(3, 6)$-regular codes to tolerate random error rate 0.01 —a factor of 5 improvement over Daskalakis et al. [2008].

In this paper we present an improvement of the error rate by another factor of 5 to 0.05, coming very close to the performance of BP. The key ingredient in our proof is a new approach to analyzing LP decoding. Instead of trying to construct a dual solution as in all the previous papers, we give a direct analysis of the primal linear program. (This also answers an open question of Feldman et al. regarding whether a primal-only analysis is possible.) At its heart, the proof relies on the fact that the LP relaxation is tight for trees. We use this to show that an LP solution can be decomposed into a distribution over codewords for every tree-like neighborhood of G so that these distributions are consistent in overlapping neighborhoods; the type of consistency that we use is inspired by hierarchies of LP relaxations, such as the Sherali–Adams hierarchy [Sherali and Adams 1990] (though all proofs are self-contained and no prior knowledge is required besides basic linear programming). We use our decomposition to define a criterion for certifying the optimality of a codeword in the right circumstances (Theorem 2), which is quite interesting on its own right. If the certificate exists, it can be found by a simple message-passing algorithm (Theorem 3), and if it exists, then LP decoding works (Theorem 4). The first such certificate was described in [Koetter and Vontobel 2006]; ours is more general and therefore occurs with high probability for much larger error rates (Theorems 1, 5). We note that prior to [Koetter and Vontobel 2006] no other analyses led to message-passing algorithms that *certify* the correctness of their answer.

As for the probability with which such a certificate exists, our calculation consists of reducing the whole problem to the study of a min-sum process on a finite tree (Definition 4), which is even amenable to analytic calculation, as done for error rate up to 0.0247 (see Section 7.1). This consists of tracing the Laplace transform of the messages exchanged by the min-sum process, as these messages move upwards on the tree. We believe that this idea of recursing the Laplace transform, rather than the density functions, of the messages is interesting on its own right and could be useful in other settings. In our setting it is rather effective in handling the min operators, which we cannot handle analytically if we trace the density functions of the messages.

Combining our analytic bounds with a MATLAB calculation, we can accommodate error rate up to 0.05 (see Section 7.2). The method seems to break down beyond 0.05, suggesting that getting to 0.084 would require new ideas. We note that our analysis does not require expansion, only high enough girth (a lower bound of $\Omega(\log \log n)$ on the girth is sufficient to make the probability of decoding error inverse polynomial in the blocklength n). Perhaps the right idea to go beyond 0.05 is to marry high-girth and expansion-based arguments, an avenue worth exploring.

An interesting byproduct of our technique is establishing that a certain reweighted version of the min-sum decoder corrects up to a 0.05-fraction of errors with high probability over the binary symmetric channel for code-rate $1/2$. To the best of our knowledge, the bound of 0.05 is the best known for re-weighted min-sum decoders over BSC for codes of rate $1/2$ (c.f. [Wiberg 1996; Frey and Koetter 2001; Chen and Fossorier 2002a; 2002b; Chen et al. 2005]). As compared to the 0.084 bound for BP, ours has the advantage that with high probability the nearest codeword can be certified to be correct.

We note that our method, being primal-only, is relatively clean —and in our opinion, easier to understand (apart maybe from the probabilistic calculation) than previous analyses. We also suspect that the idea of expressing primal solutions in terms of local tree assignments may be of wider use in applications that use LP decoding techniques and random graphs, such as in *compressed sensing.* Candès and Tao [2006] (independently of Feldman et al. though somewhat later), as part of work on *compressed sensing*, arrived at linear programming as a promising tool for a variety of reconstruction problems in compressed sensing, which include decoding random linear codes over the reals (these are *not* LDPC since the constraint graph is non-sparse). Recent work such as [Guruswami et al. 2010; Indyk 2008] makes explicit the connection between Sipser–Spielman type decoding of LDPC codes and compressed sensing using sparse matrices.

Our Main Result. All result statements including the next one assume that the message length n is some arbitrarily large constant; so for instance a girth of $\log n$ or $\log\log n$ is larger than any fixed constant.

THEOREM 1. *Let $p \leqslant 0.05$ and let $x \in \{0,1\}^n$ be a codeword of the low-density parity check code defined by a $(3,6)$-regular bipartite graph with $\Omega(\log n)$ girth. Suppose that $y \in \{0,1\}^n$ is obtained from x by flipping every bit independently with probability p. Then, with probability at least $1-\exp(-n^\gamma)$ for some constant $\gamma > 0$,*

(1) the codeword x is the unique optimal solution to the LP decoder of Feldman et al. [2005] (see LP (2) in Section 2); and

(2) a simple message-passing (dynamic programming) algorithm running in time $O(n \cdot \log^2 n)$ can find x and certify that it is the nearest codeword to y.

For LDPC codes defined by general (d_L, d_R)-regular graphs, we have the same conclusion whenever d_L, d_R, and p satisfy the conditions $p < 1 - 2^{-\frac{1}{d_R-1}}$ and

$$\sqrt{p} \quad \left(1-(1-p)^{d_R-1}\right)^{\frac{d_L-2}{2}} (1 \; - \; p)^{\frac{(d_R-1)(d_L-2)}{2}+\frac{1}{2}} \quad < \quad \frac{1}{(d_R-1)\cdot 2^{d_L-1}}\,. \qquad (1)$$

The running time is still $O(n \cdot \log^2 n)$ with the constant hidden inside the $O(\cdot)$ depending on d_L, d_R, p.

Remark 1. If we are content with a decoding success probability of $1-1/\text{poly}(n)$, then $\Omega(\log\log n)$ girth is sufficient for the results in the previous theorem. The running time of the message-passing algorithm is reduced to $O(n(\log\log n)^2)$.

2. PRELIMINARIES

Low-density parity check codes. Let G be a simple bipartite graph with bipartition (V_L, V_R), left degree d_L, and right degree d_R. Let n be the number of left vertices, and m be the number of right vertices. We will assume that V_L is the set $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For two vertices u and v of G, we let $d(u, v)$ be the distance of u and v in G. We denote by $N(u)$ the set of neighbors of u. Similarly, $N^t(u)$ is the set of vertices at distance t from u. We denote by $B(u, t)$ (for "Ball") the set of vertices at distance at most t from u.

The *parity-check code* defined by G is the set of all 0/1 assignments to the left vertices such that every right vertex has an even number of neighbors with value 1,

$$\mathcal{C}(G) \stackrel{\text{def}}{=} \{x \in \{0,1\}^n \mid \textstyle\sum_{i \in N(j)} x_i \equiv 0 \bmod 2, \text{ for all } j \in V_R\}\,.$$

The elements of $\mathcal{C}(G)$ are called *codewords*. Note that if we allow general graphs then any linear code can be realized as parity check code $\mathcal{C}(G)$ for some (generally irregular) graph G. In this paper, we will only deal with sparse (low-density) graphs, that is, the degrees d_L and d_R will be constants.

In the following, we refer to the vertices in V_L and V_R as *variable nodes* and *check nodes*, respectively.

LP Decoding. In the *nearest codeword problem* for the code $\mathcal{C}(G)$, we are given a vector $y \in \{0,1\}^n$ and the goal is to find a codeword $x \in \mathcal{C}(G)$ so as to minimize the Hamming distance $\|x - y\|_1$.

In [Feldman and Karger 2002; Feldman et al. 2005], Feldman et al. introduced the following LP relaxation for this problem:

$$\text{Minimize} \qquad \|x - y\|_1 \tag{2}$$

$$\text{subject to} \qquad x \in \bigcap_{j \in V_R} \operatorname{Conv} \mathcal{C}_j\,, \tag{3}$$

where $\operatorname{Conv} X$ denotes the convex hull of a set X of bit vectors, and $\mathcal{C}_j$ is the set of bit vectors satisfying constraint $j \in V_R$,

$$\mathcal{C}_j \stackrel{\text{def}}{=} \left\{x \in \{0,1\}^n \mid \textstyle\sum_{i \in N(j)} x_i \equiv 0 \bmod 2\right\}\,.$$

We call $x \in [0,1]^n$ an *LP solution* if it satisfies (3). We say x is an *optimal LP solution given* y if x is an LP solution that achieves the minimum distance (2) to y. An optimal LP solution can be computed in time polynomial in n. In this paper, we are interested to find conditions under which the solution of this LP coincides with the nearest codeword to y.

Before concluding this section, we note that $\|x - y\|_1$ is an affine linear function of $x \in [0,1]^n$ for any fixed $y \in \{0,1\}^n$, since $\|x - y\|_1 = \|y\|_1 + \sum_{i=1}^n (-1)^{y_i} x_i$, for all $x \in [0,1]^n$.

3. CERTIFYING THE NEAREST CODEWORD

Throughout this section and the next, $y \in \{0,1\}^n$ is the received word and $x \in \mathcal{C}(G)$ is a codeword which we are trying to certify as the nearest codeword to y. We present a certificate based on local checks that is inspired by and generalizes the key idea in the calculation of Koetter and Vontobel [2006]. The motivation for this generalization is that it allows certification/decoding in the presence of much higher noise. Our proof that this certificate works is also quite different. It is designed to easily carry over to prove that x is also the unique fractional solution to the LP.

To understand the certificate, imagine two different codewords x and x'. Whenever x and x' differ in a certain bit, say the one corresponding to $i_0 \in V_L$ then they must look very different in the neighborhood of i_0 in the constraint graph. Indeed, since x, x' differ at i_0, in order to satisfy the d_L parity constraints in $N(i_0)$ (i.e., those that are adjacent to i_0), codeword x' must actually also differ from x in some other variables that are in $N^2(i_0)$. Of course, then those other variables appear in some other constraints in $N^3(i_0)$, thus necessitating yet more variables to be flipped in $N^4(i_0)$, and so on for at least as long as the neighborhood around i_0 looks locally like a tree. (Note that the variable nodes in $B(i_0, 2T)$ have even distance to i_0 and the check nodes have odd distance.) Since the girth of the graph is large, this effect propagates quite a while, implying that x, x' differ along a large subtree rooted at i_0 in the graph.

Hence if x is not the closest codeword to y, then one could hope that this might show up locally in the neighborhood of some bit i_0: flipping this single bit in x would trigger a cascade of other bit flips, which might end up lowering the distance to y. If so, then x would be *locally suboptimal.* Note that the induced graph on $B(i_0, 2T)$, $T \leqslant \frac{1}{4}\text{girth}(G)$, is a *tree* (with degrees d_L, d_R respectively at even and odd levels), and so one can hope to find the best local perturbation from x—in the sense we will define below—efficiently via dynamic programming, and thus efficiently test for local optimality of x.

The above intuitive description raises the question: if x is globally suboptimal (i.e., not the nearest codeword to y), is it also locally suboptimal? The practical success of message-passing algorithms gives us hope that the answer may often be "yes." Such algorithms are local in the following sense: after $t < \frac{1}{4}\text{girth}(G)$ iterations, the value computed for the variable i is a "guess" for x_i given the information in the neighborhood $B(i, 2t)$; in this sense, after t rounds message passing algorithms compute a "locally optimal" solution. Several notions of "local optimality" were implicit in the algorithms of [Wiberg 1996; Frey and Koetter 2001; Chen and Fossorier 2002a; 2002b; Chen et al. 2005]. Our notion of local optimality generalizes the notions used in all of these papers, and our interest centers around showing that local optimality implies global optimality. (The reader confused by the stream of ideas coming next may wish to focus on the next two definitions and then study the simple proofs in Section 4 to understand the essence of our idea.)

Our notion of local optimality is given in Definition 2 and requires the following definition generalizing Wiberg [1996].

Definition 1. (Minimal Local Deviation) Let $T < \text{girth}(G)/4$. An assignment $\beta \in \{0,1\}^n$ is a *valid deviation of depth T at $i_0 \in V_L$* or, in short, a *T-local deviation at i_0*, if $\beta_{i_0} = 1$ and β satisfies all parity checks in $B(i_0, 2T)$, i.e.,

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

(Notice that β need not be a codeword since we do not insist that the check nodes beyond level $2T$ from i_0 are satisfied.)

A T-local deviation β at i_0 is *minimal* if $\beta_i = 0$ for every $i \notin B(i_0, 2T) \cap V_L$, and every check node j in $B(i_0, 2T)$ has at most two neighbors with value 1 in β. Note that a minimal T-local deviation at i_0 can be seen as a subtree of $B(i_0, 2T)$ of height $2T$ rooted at i_0, where every variable node has full degree and every check node has degree 2. We will refer to such trees as *skinny trees.*

An assignment $\beta \in \{0,1\}^n$ is a minimal T-local deviation if it is a minimal T-local deviation at some i_0. Note that given β there is a unique such $i_0 \overset{\text{def}}{=} \text{root}(\beta)$.

If $w = (w_1, \dots, w_T) \in [0,1]^T$ is a weight vector and β is a minimal T-local deviation, then $\beta^{(w)}$ denotes the *w-weighted* deviation

$$\beta_i^{(w)} = \begin{cases} w_t \beta_i & \text{if } d(\text{root}(\beta), i) = 2t \text{ and } 1 \leqslant t \leqslant T\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

In particular, notice that $\beta^{(w)}_{\text{root}(\beta)} = 0$. (End of Definition 1.)

For two vectors $u, v \in \{0,1\}^n$, we denote by $u \oplus v$ the coordinate-wise sum of u and v modulo 2. We extend this notation to fractional vectors in the following way: If $u \in \{0,1\}^n$ and $\bar{v} \in [0,1]^n$, then $u \oplus \bar{v} \in [0,1]^n$ denotes the vector with i^{th} coordinate $|u_i - \bar{v}_i|$. Note that, for a fixed vector $u \in \{0,1\}^n$, $u \oplus \bar{v}$ is affine linear in $\bar{v}$. Hence, for any distribution over vectors $v \in \{0,1\}^n$ and a fixed bit vector u, we have $\mathbb{E}\, u \oplus v = u \oplus (\mathbb{E}\, v)$.

The next definition tries to capture whether or not local changes to x can increase the distance from y. While our previous discussion may suggest that it suffices to consider the weight vector $w = \mathbb{1}$, [4] we will see later that the overall analysis can be improved by allowing more general weight vectors. In particular, our results on local-optimality implying global-optimality (Theorems 2 and 4) remain unaffected by the use of more general vectors, but different choices of w allow more flexibility resulting in better error rates in Theorem 5. It is still mysterious to us what the optimum choice of w is, and a theory for calculating it is left as an open problem in this paper. Intuitively, adjusting w allows us to improve the likelihood that local information in the neighborhood of variable i_0 predicts correctly the value of variable i_0. At the very least it should be clear that the vector $w = (1, 2^{-1}, \dots, 2^{1-T})$ is more "natural" than $\mathbb{1}$ since it gives equal weight to every level of a skinny tree (recall that the number of nodes in a skinny tree doubles with (half) the distance from the root). Indeed, this vector is the one we use to compute the 0.05 bound on error rate claimed in the Introduction.

Definition 2. *(Local optimality)* If $w = (w_1, \dots, w_T) \in [0,1]^T$ is a weight vector (with at least one positive coordinate) and $T < \text{girth}(G)/4$ then a codeword $x \in \{0,1\}^n$ is (T, w)-locally optimal for $y \in \{0,1\}^n$ if for all minimal T-local deviations β,

$$\|x \oplus \beta^{(w)} - y\|_1 > \|x - y\|_1\,.$$

[4] Whenever we refer to the weight vector $w = \mathbb{1}$ in this section we abuse Definition 1, by setting $\beta^{(w)}_{\text{root}(\beta)} = \beta_{\text{root}(\beta)}$ (instead of 0).

Remark: Somewhat confusingly, a globally optimal codeword does *not* necessarily have to be locally optimal (even with respect to the weight vector $w = \mathbb{1}$) because the definition of local optimality ignores what happens beyond radius T, and also restricts attention to minimal deviations.

Note that if x is locally optimal then this is some intuitive evidence that x is the nearest codeword, since changing x in just one variable i_0 seems to cause the distance from y to increase in the immediate neighborhood of i_0. Koetter and Vontobel [2006] make this intuition precise for $w = \mathbb{1}$, in which case they show that a locally optimal x is also globally optimal, that is, the nearest codeword to y. (In fact, all previous theoretical analyses of message passing algorithms have some notion of local optimality but none of them were known to imply global optimality.) Our proof works for general w.

THEOREM 2. *Let $T < \frac{1}{4}\operatorname{girth}(G)$ and $w = (w_1, \ldots, w_T)$ be any non-negative weight vector with at least one positive coordinate.*

(1) If x is a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$, then x is also the unique nearest codeword for y.

(2) Moreover, given w, x, and y a simple dynamic programming algorithm running in time $O(d_L d_R |w| \cdot T \cdot n \log n)$ can certify that x is a (T, w)-locally optimal codeword for y, where $|w|$ is the number of bits required to represent a coordinate of the weight vector w.

Observe that since the local neighborhood at each node i_0 is tree-like, the minimal T-local deviation β at i_0 that minimizes $\|x \oplus \beta^{(w)} - y\|_1$, for a given pair of x and y, can be computed by a simple dynamic programing algorithm in time near-linear in the size of the neighborhood. Theorem 2(b) shows that it is possible to interleave these computations for all possible i_0's achieving an overall running time of $O(T \cdot n \log n)$ (which is near-linear since we are going to choose $T = O(\log n)$ or $T = O(\log \log n)$ later in the proof). The details of the dynamic program are presented in Section 4. (The running times quoted in this paragraph are hiding the dependence on the degrees d_L, d_R and the description complexity of vector w. For the precise running time see the statement of Theorem 2.)

Theorem 2 implies that we can certify that x is the nearest codeword for y by verifying the local optimality condition. It raises two questions:

(1) How can we find a locally optimal codeword if it exists?

(2) What is the chance that the nearest codeword satisfies the local optimality condition?

The first question has been studied in the context of message-passing algorithms. For $w = \mathbb{1}$, Wiberg [1996] showed that the well-known min-sum decoding algorithm can find locally optimal codewords. Other specific weight functions were suggested and analyzed in several works [Frey and Koetter 2001; Chen and Fossorier 2002a; 2002b; Chen et al. 2005]. We show how to make min-sum decoding work for arbitrary weight vectors (which allows much more freedom in deriving analytic error bounds). The proof of the following theorem and details of the algorithm are presented in Section 5.

Theorem 3. *Let $T < \frac{1}{4}\operatorname{girth}(G)$ and $w = (w_1, \ldots, w_T)$ be a non-negative weight vector with at least one positive coordinate. Suppose that x is a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$. Then the w-reweighted min-sum algorithm on input y computes x in T iterations. The running time of the algorithm is $O(d_L d_R |w| \cdot T \cdot n \log n)$, where $|w|$ is the number of bits required to represent a coordinate of the weight vector w.*

Since the focus of this paper is on LP decoding, we next address whether LP decoding can find locally optimal solutions. To this end we extend Theorem 2 in order to show that a locally optimal solution is not only the nearest codeword to y but also the unique optimal LP solution given y. For the case $w = \mathbb{1}$, this was also established by Koetter and Vontobel [2006].

Theorem 4. *Let $T < \frac{1}{4}\operatorname{girth}(G)$ and $w = (w_1, \ldots, w_T)$ be any non-negative weight vector with at least one positive coordinate. Suppose that x is a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$. Then x is also the unique optimal LP solution given y.*

The proof for $w = \mathbb{1}$ in [Koetter and Vontobel 2006] proceeded by constructing an appropriate dual solution in an iterative manner. Our proof in Section 6 yields a more general result, and is completely different, since it only looks at the primal LP.

Now we address the second question mentioned above, regarding the probability that the nearest codeword is locally optimal. (Showing higher probabilities for this event was the main motivation for introducing general weight vectors w.) We prove the following theorem in Section 7. The theorem asserts the existence of certain weight vectors. Their explicit choices are described in the proof. We just note here that each coordinate of these vectors may only be a function of d_L, d_R and p, and in particular does not depend on n. We also note that we did not address the question of numerically computing/approximating these weight vectors. Their computation is irrelevant as far as proving the success of the Linear Programming decoder goes, but is needed if one chooses to instead use the w-reweighted min-sum decoder of Theorem 3, or wants to certify the optimality of the codeword output by the decoder, using the algorithm of Theorem 2.

Theorem 5. *Let G be a (d_L, d_R)-regular bipartite graph and $T < \frac{1}{4}\operatorname{girth}(G)$. Let also $p \in (0,1)$ and $x \in \{0,1\}^n$ be a codeword in $\mathcal{C}(G)$. Finally, suppose that y is obtained from x by flipping every bit independently with probability p.*

(1) *If d_L, d_R, and p satisfy the condition*

$$\min_{t>0}\Big\{ \big((1-p)\ e^{-t} + p\ e^{t}\big) \cdot \big((1-p)^{d_R-1}\ e^{-t} + (1-(1-p)^{d_R-1})\ e^{t}\big)^{d_L-2} \Big\} < \frac{1}{d_R-1}, \quad (4)$$

then x is $(T, \mathbb{1})$-locally optimal with probability at least $1 - n \cdot c^{1-(d_L-1)^{T-1}}$ for some $c > 1$. For $(d_L, d_R) = (3,6)$, Condition (4) is satisfied whenever $p \leqslant 0.02$.

(*2*) *If* $d_L, d_R \geqslant 2$ *and* p *satisfy the conditions* $p < 1 - 2^{-\frac{1}{d_R - 1}}$ *and*

$$\sqrt{p}\ \left(1 - (1-p)^{d_R - 1}\right)^{\frac{d_L - 2}{2}} (1 - p)^{\frac{(d_R-1)(d_L-2)}{2} + \frac{1}{2}} \ < \ \frac{1}{(d_R - 1) \cdot 2^{d_L - 1}}, \quad (5)$$

then there exists a weight vector $w \in [0,1]^T$ *such that* x *is* (T, w)*-locally optimal with probability at least* $1 - n \cdot c' \cdot c^{1 - (d_L - 1)^{T-1}}$ *for some constants* c' *and* $c > 1$. *For* $(d_L, d_R) = (3, 6)$*, Condition (5) is satisfied whenever* $p \leqslant 0.0247$.

(*3*) *There exists a weight vector* w *such that, if* $(d_L, d_R) = (3, 6)$ *and* $p \leqslant 0.05$*, then* x *is* (T, w)*-locally optimal with probability at least* $1 - n \cdot c' \cdot c^{-2^{T-16}}$ *for some constants* c' *and* $c > 1$.

Given Theorems 2, 3, 4, and 5 we can obtain the proof of Theorem 1 as follows. Suppose that $n = \Omega_{p,d_L,d_R}(1)$ is a large enough constant and take $T = \Theta(\log n) < \frac{1}{4}\mathrm{girth}(G)$. From Theorem 5, there exists a weight vector w such that, with probability at least $1 - \exp(-n^\gamma)$ for some constant γ depending linearly on the leading constant in front of $\log n$ in the choice of T (as well as on p, d_L, d_R), the codeword x is (T, w)-locally optimal. From Theorem 4 it follows then that x is the unique optimal LP solution given y. Also, from Theorem 3, it follows that x can be found by the w-reweighted min-sum algorithm in $T = O(\log n)$ iterations, so time $O_{d_R,d_L,p}(n \log^2 n)$ overall, and by Theorem 2 it can be certified that x is the nearest codeword to y. (If the girth is $\Omega(\log \log n)$ then the decoding still succeeds with probability $1 - 1/\mathrm{poly}(n)$.)

4. LOCAL OPTIMALITY IMPLIES GLOBAL OPTIMALITY

Proof of Theorem 2. In this section $y \in \{0,1\}^n$ is the received word, $x \in \{0,1\}^n$ is a locally optimal codeword in $\mathcal{C}(G)$, and $x' \in \{0,1\}^n$ is a codeword different from x. We wish to show $\|x' - y\|_1 > \|x - y\|_1$.

The following lemma is the key to our proof of Theorem 2, and may be viewed as one of our key new contributions.

LEMMA 1. *Let* $T < \frac{1}{4}\mathrm{girth}(G)$*. Then, for every codeword* $z \neq 0$*, there exists a distribution over minimal* T*-local deviations* β *such that, for every weight vector* $w \in [0,1]^T$,

$$\mathbb{E}\, \beta^{(w)} = \alpha z\,,$$

where $\alpha \in [0,1]$ *is some scaling factor depending upon* w.

Before proving the lemma, let us first see how we can finish the proof of Theorem 2 using such a distribution over minimal local deviations. This proof (as well as the related proof of Theorem 4) is at the heart of our paper.

PROOF OF THEOREM 2. Let x be a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$. We want to show that for every codeword $x' \neq x$, the distance to y increases, that is, $\|x - y\|_1 < \|x' - y\|_1$. The main idea is to observe that $z = x \oplus x'$ is also a codeword, and hence by Lemma 1, there exists a distribution over minimal T-local deviations β such that $\mathbb{E}\,\beta^{(w)} = \alpha z$ for the codeword $z = x \oplus x'$. Now it is easy to complete the proof using local optimality of x. Let $f\colon [0,1]^n \to \mathbb{R}$ be the affine linear function $f(u) = \|x \oplus u - y\|_1 = \|x - y\|_1 + \sum_{i=1}^n (-1)^{x_i + y_i} u_i$. Now,

$$\begin{aligned} \|x - y\|_1 &< \mathbb{E}\|x \oplus \beta^{(w)} - y\|_1 \qquad \text{(by local optimality of } x\text{)} \\ &= \|x \oplus (\alpha z) - y\|_1 \qquad \text{(affine linearity of } f\text{)} \\ &= \alpha\|x' - y\|_1 + (1 - \alpha)\|x - y\|_1 \quad \text{(aff. lin. of } f\text{)}\,, \end{aligned}$$

which implies $\|x - y\|_1 < \|x' - y\|_1$ as desired.

To conclude the proof of the theorem we argue that certifying that a given x is a (T, w)-locally optimal codeword for a given $y \in \{0,1\}^n$ can be carried out in $O(d_L d_R |w| \cdot T \cdot n \log n)$ time with dynamic programming, where $|w|$ is the number of bits required to represent each coordinate of the weight vector w. The algorithm proceeds in T iterations, where each iteration comprises two phases. In the first phase of iteration $t = 1, \ldots, T$, for every pair of adjacent variable and check nodes $(u, v) \in E(G) \cap V_L \times V_R$, variable node u sends a message $\mu^t_{u \to v} \in \mathbb{R}^{\{0,1\}}$ to check node v, while in the second phase of the iteration a message $\mu^t_{v \to u} \in \mathbb{R}^{\{0,1\}}$ is sent in the opposite direction of the edge. The messages are as follows

(1) *Initialization:* For all $(u, v) \in V_L \times V_R \cap E(G)$, $z_u \in \{0,1\}$:

$$\mu^1_{u \to v}(z_u) = |y_u - |x_u - w_T \cdot z_u||.$$

(2) *Variable-to-check node message:* For all $(u, v) \in V_L \times V_R \cap E(G)$, $z_u \in \{0,1\}$, $t > 1$:

$$\mu^t_{u \to v}(z_u) = |y_u - |x_u - w_{T-t+1} z_u|| + \sum_{v' \in N(u) \setminus \{v\}} \mu^{t-1}_{v' \to u}(z_u).$$

(3) *Check-to-variable node message:* For all $(v, u) \in V_R \times V_L \cap E(G)$, $t \geqslant 1$:

$$\begin{aligned} \mu^t_{v \to u}(0) &= \sum_{u' \in N(v) \setminus \{u\}} \mu^t_{u' \to v}(0); \\ \mu^t_{v \to u}(1) &= \min_{x \in \{0,1\}^{N(v) \setminus \{u\}}:\ ||x||_1 \leqslant 1} \ \sum_{u' \in N(v) \setminus \{u\}} \mu^t_{u' \to v}(x_{u'}). \end{aligned}$$

(4) *Termination:* After iteration T is completed every variable node $u \in V_L$ computes:

$$\mu_u = |y_{i_0}| + \sum_{v \in N(i_0)} \mu^T_{v \to i_0}(1).$$

It is easy to see that, for all $u \in V_L$, the message μ_u computed by the algorithm described above equals the minimum $\|x \oplus \beta^{(w)} - y\|_1$ over all minimal T-local deviations β at u. Hence, comparing the computed μ_u's against $\|x - y\|_1$ will certify correctly the local optimality of x. The running time of the algorithm is $O(d_L d_R |w| \cdot T \cdot n \log n)$, where $|w|$ is the number of bits required to represent each coordinate of the weight vector w. Indeed, there are $O(d_L n T)$ messages computed in the course of the algorithm (since there are $O(T)$ iterations and there are $O(d_L n)$ messages computed in each iteration), each message has size $O(|w| \log n)$, and takes time $O(d_R |w| \log n)$ to compute. □

4.1 Proof of Lemma 1

Constructing Distributions over Minimal Local Deviations for Codewords. Let $z \in \{0,1\}^n$ be a codeword. We want to construct a distribution over minimal local deviations such that the mean of the distribution (viewed as a vector in $\Re^n$) is proportional to z.

For every variable node $i \in V_L$ with $z_i \neq 0$, we define a distribution over subtrees τ_i of G of height $2T$ rooted at i: The idea is that we grow τ_i minimally and randomly inside the non-zeros of z starting from the variable node i. Consider the neighborhood $B(i, 2T)$ and direct the edges away from the root i. Remove all variable nodes in this neighborhood with value 0 in z. Remove now the vertices that are no longer reachable from i. This leaves a tree (rooted at i) in which the degree is a nonzero even number at each check node and d_L at each variable node. In this remaining tree, pick a random subtree τ_i with full out-degree at variable nodes and out-degree 1 at check nodes.

Suppose now that we choose such a tree τ_i for all i with $z_i \neq 0$, such that these trees are mutually independent. Independently from the choices of the trees, we also choose i_0 uniformly at random from the support of z (that is, we pick i_0 with probability $z_{i_0}/\|z\|_1$), and define β as

$$\beta_i = \begin{cases} 1 & \text{if } i \in \tau_{i_0}\,, \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $\mathbb{P}$ the joint probability measure over the trees $\{\tau_i\}_{i:z_i \neq 0}$, the variable node i_0 and the assignment β. Before concluding the proof of Lemma 1, we make a few observations about the random subtrees τ_i. First, the number of nodes at level $2t \leqslant 2T$ of any tree τ_i is always exactly $d_L(d_L-1)^{t-1}$ (the root has out-degree d_L). Second, for any two variable nodes i, i' with $z_i = z_{i'} = 1$ the above process treats them symmetrically:

$$\mathbb{P}\left\{i' \in \tau_i\right\} = \mathbb{P}\left\{i \in \tau_{i'}\right\}. \tag{6}$$

Indeed, if $d(i, i') > 2T$, then both of the probabilities are 0 (since the height of the trees is $2T$). Otherwise, the nodes i, i' are connected by a unique path of length $\leqslant 2T$, say $(i, j_0, i_1, j_1, \ldots, i_{t-1}, j_{t-1}, i')$. If there exists some variable node i_ℓ, $\ell \in \{1, \ldots, t-1\}$, in this path with $z_{i_\ell} = 0$ then both of the probabilities are zero. Otherwise, let $d_r = \sum_{i \in N(j_r)} z_i$ be the number of neighbors of j_r that are in the support of z. Then both of the probabilities in (6) are equal to

$$\frac{1}{(d_0 - 1) \cdots (d_{t-1} - 1)}\,,$$

which completes the proof of the claim in (6).

Armed with these observations, we can analyze our distribution over minimal local deviations β: If $z_i = 0$, then $\beta_i = 0$ with probability 1. Hence, we may assume $z_i = 1$. Then,

$$\begin{aligned}
\mathbb{E}\,\beta_i^{(w)} &= \sum_{t=1}^{T} w_t \sum_{\substack{i' \in N^{2t}(i) \\ z_{i'}=1}} \mathbb{P}\left\{i_0 = i'\right\} \mathbb{P}\left\{i \in \tau_{i'}\right\} \\
&\overset{(6)}{=} \sum_{t=1}^{T} w_t \sum_{\substack{i' \in N^{2t}(i) \\ z_{i'}=1}} \tfrac{1}{\|z\|_1} \mathbb{P}\left\{i' \in \tau_i\right\} \\
&= \tfrac{1}{\|z\|_1} \sum_{t=1}^{T} w_t \,\mathbb{E}\left|\tau_i \cap N^{2t}(i)\right| \\
&= \tfrac{1}{\|z\|_1} \sum_{t=1}^{T} w_t \cdot d_L (d_L - 1)^{t-1}
\end{aligned}$$

Therefore, we have the desired conclusion $\mathbb{E}\,\beta^{(w)} = \alpha z$ with $\alpha = \sum_{t=1}^{T} w_t \cdot d_L(d_L - 1)^{t-1}/\|z\|_1$. $\square$

5. FINDING LOCALLY OPTIMAL CODEWORDS

We briefly describe how to find a (T, w)-locally optimal codeword for a given $y \in \{0,1\}^n$ if such a codeword exists. The algorithm is a weighted version of the min-sum algorithm (see, e.g., Chapter 3 of [Wiberg 1996]). It proceeds in T iterations, $T < \frac{1}{4}\mathrm{girth}(G)$, where each iteration comprises two phases. In the first phase of iteration $t = 1, \ldots, T$, for every pair of adjacent variable and check nodes $(u, v) \in E(G) \cap V_L \times V_R$, variable node u sends a message $\mu^t_{u \to v} \in \mathbb{R}^{\{0,1\}}$ to check node v, while in the second phase of the iteration a message $\mu^t_{v \to u} \in \mathbb{R}^{\{0,1\}}$ is sent in the opposite direction of the edge. The messages are as follows

(1) *Initialization:* For all $(u, v) \in E(G) \cap V_L \times V_R$, $z_u \in \{0, 1\}$:

$$\mu^1_{u \to v}(z_u) = |w_T \cdot z_u - y_u|.$$

(2) *Variable-to-check node message:* For all $(u, v) \in E(G) \cap V_L \times V_R$, $z_u \in \{0, 1\}$, $t > 1$:

$$\mu^t_{u \to v}(z_u) = |w_{T-t+1} \cdot z_u - y_u| + \sum_{v' \in N(u) \setminus \{v\}} \mu^{t-1}_{v' \to u}(z_u).$$

(3) *Check-to-variable node message:* For all $(v, u) \in E(G) \cap V_R \times V_L$, $z_u \in \{0, 1\}$, $t \geqslant 1$:

$$\mu^t_{v \to u}(z_u) = \min_{\substack{x \,:\, \sum_{u' \in N(v)} x_{u'} \equiv 0 \bmod 2 \\ \text{and } x_u = z_u}} \sum_{u' \in N(v) \setminus \{u\}} \mu^t_{u' \to v}(x_{u'}).$$

(4) *Termination:* After iteration T is completed, every variable node $u \in V_L$ computes:

$$z_u^T = \arg\min_{x_u \in \{0,1\}} \Big\{ y_u + \sum_{v' \in N(u)} \mu^T_{v' \to u}(x_u) \Big\}.$$

We show the following lemma.

LEMMA 2. *If x is a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$ and $T < \frac{1}{4}\mathrm{girth}(G)$, then for all u the values computed at termination by the above algorithm satisfy:*

$$z_u^T = x_u.$$

PROOF. Due to the symmetry of the code we can assume, without loss of generality, that $x = (0, 0, \ldots, 0)$. Hence, we want to establish that, for all u, $z_u^T = 0$. For $\chi \in \{0,1\}^n$ let us define:

$$\chi_i^{(w)} = \begin{cases} w_t \chi_i & \text{if } d(u,i) = 2t \text{ and } 1 \leqslant t \leqslant T\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

It is clear from the definition of our algorithm that, for every variable node u and because $T < \frac{1}{4}\mathrm{girth}(G)$,

$$z_u^T = \arg \min_{\substack{\chi \,\in\, \{0,1\}^n\colon\ \chi \text{ satisfies all} \\ \text{check nodes in } B(u,2T)}} \Big\{ ||\chi^{(w)} - y||_1 \Big\}. \tag{7}$$

Now we use the following decomposition lemma due to Wiberg (see Lemma 4.3 in [Wiberg 1996]).

LEMMA 3. *For $s \in \{0,1\}$ let*

$$\mathcal{B}^s := \{\chi \in \{0,1\}^n \mid \chi \textit{ satisfies all check nodes in } B(u,2T) \textit{ and } \chi_u = s\}.$$

Moreover, let $\mathcal{E}$ be the set of minimal T-local deviations at u. Then any $\chi \in \mathcal{B}^1$ can be decomposed as $\chi = \chi' + e$, where $\chi' \in \mathcal{B}^0$, $e \in \mathcal{E}$ and χ', e have disjoint supports (i.e, for all $u' \in V_L$, $\chi'_{u'}$ and $e_{u'}$ cannot be both 1).

Armed with this lemma we can conclude the proof. Indeed, assume that $z_u^T = 1$. Then (7) implies that there exists some $\hat{\chi} \in \mathcal{B}^1$ such that

$$||\hat{\chi}^{(w)} - y||_1 \leqslant ||\chi'^{(w)} - y||_1, \forall \chi' \in \mathcal{B}^0,$$

where $\mathcal{B}^0$ and $\mathcal{B}^1$ are defined as in Lemma 3. But the same lemma implies that

$$\hat{\chi} = \chi' + e,$$

where $\chi' \in \mathcal{B}^0$, $e \in \mathcal{E}$ and χ', e have disjoint support. It is easy to check that

$$||\hat{\chi}^{(w)} - y||_1 = ||\chi'^{(w)} - y||_1 + ||e^{(w)} - y||_1 - ||y||_1.$$

Combining the last three equations we have that

$$||e^{(w)} - y||_1 \leqslant ||y||_1,$$

a contradiction to our assumption that 0^n is (T, w)-locally optimal for y at u. □

Hence, the algorithm described above computes a (T, w)-locally optimal codeword for a given $y \in \{0,1\}^n$, if such a codeword exists. The running time of the algorithm is $O(d_L d_R n \cdot |w| \cdot T^2 \cdot \log(d_L d_R))$, where $|w|$ is the number of bits required to represent each coordinate of the weight vector w. Indeed, there are $O(d_L n T)$

messages computed in the course of the algorithm (since there are $O(T)$ iterations and there are $O(d_L n)$ messages computed in each iteration), each message has bit length $O(|w| \cdot T \cdot \log(d_L d_R))$, and takes time $O(d_R \cdot |w| \cdot T \cdot \log(d_L d_R))$ to compute. To argue the latter we use a straightforward dynamic program to do the minimization of Step (3) in $O(d_R \cdot |w| \cdot T \cdot \log(d_L d_R))$ bit operations.

6. LOCAL OPTIMALITY IMPLIES LP OPTIMALITY

Proof of Theorem 4. Let $x \in \{0,1\}^n$ be a codeword in $\mathcal{C}(G)$ and let $y \in \{0,1\}^n$. The following lemma is completely analogous to Lemma 1, and follows from the fact that LP solutions look locally like distributions over codewords.

Lemma 4. *Let $T < \frac{1}{4}\,\mathrm{girth}(G)$ and $w \in [0,1]^T$. Then for every non-zero LP solution $z \in [0,1]^n$, there exists a distribution over minimal T-local deviations β such that*

$$\mathbb{E}\, \beta^{(w)} = \alpha z\,,$$

where $\alpha \overset{\mathrm{def}}{=} \sum_{t=1}^{T} w_t \frac{d_L (d_L - 1)^{t-1}}{\|z\|_1}$.

Using a distribution over minimal local deviations from Lemma 4, we can prove Theorem 4 in almost the same way as Theorem 2 in the previous section. The only additional ingredient is the following simple property of LP solutions. Recall that for $x \in \{0,1\}^n$ and $x' \in [0,1]^n$, we denote by $x \oplus x'$ the vector whose ith coordinate is $|x_i - x'_i|$. The next lemma is straightforward using the observation that the defining property of an LP solution (specifically, (3)) is that locally (for every check node) it can be viewed as a convex combination of even-parity vectors.

Lemma 5. *Let x be a codeword and x' be an LP solution (i.e., it satisfies (3)). Then $x \oplus x'$ is also an LP solution.*

Now we can prove the main theorem.

Proof of Theorem 4. Let x be a (T, w)-locally optimal codeword for $y \in \{0,1\}^n$. We want to show that for every LP solution $x' \neq x$, the distance to y increases, that is, $\|x - y\|_1 < \|x' - y\|_1$. By Lemma 4, there exists a distribution over minimal T-local deviations β such that $\mathbb{E}\, \beta^{(w)} = \alpha z$ for the LP solution $z = x \oplus x'$. Let $f\colon [0,1]^n \to \mathbb{R}$ be the affine linear function $f(u) = \|x \oplus u - y\|_1 = \|x - y\|_1 + \sum_{i=1}^{n} (-1)^{x_i + y_i} u_i$. Now,

$$\begin{aligned} \|x - y\|_1 &< \mathbb{E} \|x \oplus \beta^{(w)} - y\|_1 && \text{(by local optimality of } x\text{)} \\ &= \|x \oplus (\alpha z) - y\|_1 && \text{(affine linearity of } f\text{)} \\ &= \alpha \|x' - y\|_1 + (1 - \alpha)\|x - y\|_1 && \text{(aff. lin. of } f\text{)}\,, \end{aligned}$$

which implies $\|x - y\|_1 < \|x' - y\|_1$ as desired. □

6.1 Proof of Lemma 4

Constructing Distributions over Minimal Local Deviations for LP Solutions. Let $z \in [0,1]^n$ be a non-zero LP solution. The proof of the current lemma is very similar to the proof of Lemma 1 (the integral case). The following lemma is essentially the only additional ingredient of the proof.

Lemma 6. *For every nonzero LP solution z and every $j \in V_R$, we can find a function $\rho_j \colon V_L \times V_L \to \mathbb{R}_+$ such that*

(1) for every neighbor $i \in N(j)$

$$z_i = \sum_{i' \in N(j) \setminus \{i\}} \rho_j(i, i')\,,$$

(2) for any two neighbors $i, i' \in N(j)$,

$$\rho_j(i, i') = \rho_j(i', i)\,.$$

Proof. Since z is an LP solution, it is a convex combination of assignments in $\mathcal{C}_j = \{\gamma \in \{0,1\}^n \mid \sum_{i \in N(j)} \gamma_i \equiv 0 \bmod 2\}$. Hence, there are multipliers $\alpha_\gamma \geqslant 0$ with $\sum_{\gamma \in \mathcal{C}_j} \alpha_\gamma = 1$ such that $z = \sum_{\gamma \in \mathcal{C}_j} \alpha_\gamma \gamma$. Now we can define $\rho_j(i, i')$ as

$$\rho_j(i, i') = \sum_{\substack{\gamma \in \mathcal{C}_j \\ \gamma_i = 1, \gamma_{i'} = 1}} \alpha_\gamma \frac{\gamma_i \gamma_{i'}}{\sum_{i'' \in N(j) \setminus \{i\}} \gamma_{i''}}\,.$$

The second property (symmetry) follows from the fact $\sum_{i'' \in N(j) \setminus \{i\}} \gamma_{i''} = \sum_{i'' \in N(j) \setminus \{i'\}} \gamma_{i''}$ for $\gamma_i = \gamma_{i'} = 1$. The first property (marginalization) can be verified directly. □

Remark 2. The function ρ_j has a natural probabilistic interpretation. As in the proof, we can think of z as the mean of a distribution over assignments $\gamma \in \mathcal{C}_j$. For a variable node $i \in N(j)$ with $z_i > 0$, we sample an assignment γ from this distribution conditioned on the event $\gamma_i = 1$. Now we output a random variable node $i' \in N(j) \setminus \{i\}$ with $\gamma_{i'} = 1$. The probability that we output i' is exactly $\rho_j(i, i')/z_i$. Note that the function ρ_j is not fully determined by z, since we could realize z as mean of two very different distributions.

The distribution over minimal T-local deviations β we construct for the LP solution z is very similar to the distribution used in Lemma 1 (especially taking into account the probabilistic interpretation of the functions ρ_j). As before, we first define for every variable node i with $z_i > 0$, a distribution over height-$2T$ skinny trees τ_i rooted at i: we start by choosing i as the root. From any chosen variable node, we branch to all its neighbors in the next level. From any chosen check node j, we go to a random neighbor, chosen according to the transition probabilities $\rho_j(\cdot, \cdot)$. More precisely, if we reached j from a variable node i_{in}, then we select the next variable node i_{out} at random from $N(j) \setminus \{i_{\text{in}}\}$, with probability $\rho_j(i_{\text{in}}, i_{\text{out}})/z_{i_{\text{in}}}$.

We can now define the distribution over minimal local deviations β: For every variable node i with $z_i > 0$, we independently choose a tree τ_i as described above. Independently from the choices of the trees, we also choose a variable node i_0 at random according to the probabilities $z_{i_0}/\|z\|_1$. Finally, we output the minimal T-local deviation β defined by

$$\beta_i = \begin{cases} 1 & \text{if } i \in \tau_{i_0}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

We make a few observations. First, the number of nodes at level $2t$ of τ_{i_0} is exactly $d_L(d_L - 1)^{t-1}$. Second, for any two variable nodes i, i' that lie in the support of z

and have distance $\leqslant 2T$ to each other, the above process for constructing a random skinny tree treats them symmetrically:

$$z_i \mathbb{P}\left\{i' \in \tau_i\right\} = z_{i'} \mathbb{P}\left\{i \in \tau_{i'}\right\} \tag{8}$$

The reason is that i, i' are connected by a single path of length $\leqslant 2T$, say $(i, j_0, i_1, j_1, \ldots, i_{t-1}, j_{t-1}, i')$. If $z_{i_\ell} = 0$ for some variable node i_ℓ on this path, both sides of (8) are naught. Otherwise, both sides of (8) are equal to

$$\frac{\rho_{j_0}(i, i_1) \cdots \rho_{j_{t-1}}(i_{t-1}, i')}{z_{i_1} \cdots z_{i_{t-1}}}\,.$$

Armed with these observations, we can compute the mean of our distribution over (w-weighted) minimal local deviations: For every i with $z_i > 0$, we have

$$\begin{aligned}
\mathbb{E}\,\beta_i^{(w)} &= \sum_{t=1}^{T} w_t \sum_{\substack{i' \in N^{2t}(i) \\ z_{i'} > 0}} \mathbb{P}\left\{i_0 = i'\right\} \mathbb{P}\left\{i \in \tau_{i'}\right\} \\
&= \sum_{t=1}^{T} w_t \sum_{\substack{i' \in N^{2t}(i) \\ z_{i'} > 0}} \tfrac{z_{i'}}{\|z\|_1} \mathbb{P}\left\{i \in \tau_{i'}\right\} \\
&\overset{(8)}{=} \sum_{t=1}^{T} w_t \sum_{\substack{i' \in N^{2t}(i) \\ z_{i'} > 0}} \tfrac{z_i}{\|z\|_1} \mathbb{P}\left\{i' \in \tau_i\right\} \\
&= z_i \cdot \tfrac{1}{\|z\|_1} \sum_{t=1}^{T} w_t\, \mathbb{E}\left|\tau_i \cap N^{2t}(i)\right| \\
&= z_i \cdot \tfrac{1}{\|z\|_1} \sum_{t=1}^{T} w_t \cdot d_L (d_L - 1)^{t-1}
\end{aligned}$$

Therefore, we have the desired conclusion $\mathbb{E}\,\beta^{(w)} = \alpha z$ with $\alpha = \sum_{t=1}^{T} w_t \cdot d_L (d_L - 1)^{t-1} / \|z\|_1$. □

7. PROBABILISTIC ANALYSIS OF LOCAL OPTIMALITY ON TREES

For the purposes of this section, let us define a notion of optimality of a codeword in the immediate neighborhood of a variable node $i_0 \in V_L$, by appropriately restricting Definition 2 of Section 3.

Definition 3. *(Single Neighborhood Optimality)* A codeword $x \in \{0,1\}^n$ is (i_0, T, w)-locally optimal for $y \in \{0,1\}^n$ if for all minimal T-local deviations β at i_0,

$$\|x \oplus \beta^{(w)} - y\|_1 > \|x - y\|_1\,.$$

Now, for a fixed weight vector w, variable node $i_0 \in V_L$, and codeword $x \in \mathcal{C}(G)$, we are interested in the probability

$$\mathbb{P}_{y \sim_p x} \left\{ \begin{array}{c} x \text{ is} \\ (i_0, T, w)\text{-locally} \\ \text{optimal for } y \end{array} \right\}, \tag{9}$$

where $\mathbb{P}_{y\sim_p x}$ is the measure defined by flipping every bit of x independently with probability p to obtain y.

If $T < \frac{1}{4}$ girth(G), the symmetry of the code and the channel imply that Probability (9) does not depend on the choice of x and i_0. Therefore, estimating this probability can be reduced to the study of a concrete random process on a fixed regular tree (see Definition 4).

Definition 4. *(T,ω)-Process on a (d_L,d_R)-Tree:* Let $\mathcal{T}$ be a directed tree of height $2T$, rooted at a vertex v_0. The root has out-degree d_L and the vertices in level $2T$ have out-degree 0. The vertices in any other even level have out-degree $d_L - 1$. The vertices in odd levels have out-degree $d_R - 1$. The vertices in even levels are called *variable nodes* and the vertices in odd levels are called *check nodes*. For $\ell \in \{0,\dots,2T\}$, let us denote by V_ℓ the set of vertices of $\mathcal{T}$ at height ℓ (the leaves have height 0 and the root has height $2T$).
A *skinny subtree* of $\mathcal{T}$ is a vertex set $\tau \subseteq V(\mathcal{T})$ such that the induced subgraph is a directed connected tree rooted at v_0 where each variable node in τ has full out-degree and each check node in τ has out-degree exactly 1. For a $\{1,-1\}$-assignment η to the variable nodes of $\mathcal{T}$ and $\omega \in [0,1]^{\{0,1,\dots,T-1\}}$, we define the ω-weighted value of a skinny subtree τ as

$$\mathrm{val}_\omega(\tau;\eta) \overset{\mathrm{def}}{=} \sum_{\ell=0}^{T-1} \sum_{v\in\tau\cap V_{2\ell}} \omega_\ell \cdot \eta_v \,.$$

(In words, we sum the values of the variable nodes in τ weighted according to their height.)
For $p \in (0,1)$, we are interested in the probability

$$\mathbb{P}_p\left\{\min_\tau \mathrm{val}_\omega(\tau;\eta) > 0\right\}\,,$$

where the minimum is over all skinny subtrees τ and the measure $\mathbb{P}_p$ on η is defined by choosing $\eta_v = 1$ with probability $1-p$, and $\eta_v = -1$ with probability p.

Notation: Notice the subtle difference between ω and w. Both ω and w represent weight vectors. However, ω, used in Definition 4 and throughout this section, assigns weights to the levels of a tree in a bottom-up fashion, i.e. its first coordinate corresponds to the leaves of the tree. On the other hand, w, used throughout the previous sections, assigns weights to the levels of a tree in a top-down fashion, i.e. its first coordinate corresponds to the variables closest to the root of the tree.

The following lemma makes the connection of the above random process to local optimality and the relation of ω and w precise.

Lemma 7. *Let $T < \frac{1}{4}$girth(G).*

$$\mathbb{P}_{y\sim_p x}\left\{\begin{array}{c} x \textit{ is } (i_0,T,w)\textit{-locally} \\ \textit{optimal for } y \end{array}\right\} \qquad = \qquad \mathbb{P}_p\left\{\min_\tau \mathrm{val}_\omega(\tau;\eta) > 0\right\},$$

where $\omega_\ell = w_{T-\ell}$, for $\ell = 0,\dots,T-1$.

Proof. The subgraph of G in $B(i_0,2T)$ is isomorphic to the tree $\mathcal{T}$ of Definition 4. Let then $\varphi\colon B(i_0,2T) \to V(\mathcal{T})$ be one of the isomorphisms between the two graphs.

First, we observe that x is (i_0,T,w)-locally optimal for y if and only if

$$\min_\beta \sum_{t=1}^{T} \sum_{i\in N^{2t}(i_0)} w_t \cdot (-1)^{x_i+y_i}\beta_i > 0\,,$$

where the minimum is over all minimal T-local deviations β at i_0. The reason is that $\|x \oplus \beta^{(w)} - y\|_1 - \|x - y\|_1$ can be expanded as $\sum_{i=1}^{n} \beta_i^{(w)} (-1)^{x_i + y_i}$, as is easily checked.

We also note that the isomorphism φ gives rise to a bijection between the minimal deviations β in $B(i_0, 2T)$ and the skinny subtrees τ of $\mathcal{T}$. We define $\varphi(\beta)$ to be the skinny tree that contains all variable nodes v with $\beta_{\varphi^{-1}(v)} = 1$.

Now imagine coupling the random variables y and η in such a way that

$$\eta_v = (-1)^{x_i + y_i}\,, \text{ where } i = \varphi^{-1}(v)\,.$$

These η_v's are iid $+1/-1$ distributed with probability p of being -1, so the distribution is correct. Furthermore we claim that $\|x \oplus \beta^{(w)} - y\|_1 - \|x - y\|_1 = \mathrm{val}_\omega(\varphi(\beta); \eta)$. The reason is that for all β and $\tau = \varphi(\beta)$

$$\begin{aligned}
&\|x \oplus \beta^{(w)} - y\|_1 - \|x - y\|_1 \\
&\qquad = \sum_{t=1}^{T} \sum_{i \in N^{2t}(i_0)} w_t \cdot (-1)^{x_i + y_i} \beta_i = \sum_{t=1}^{T} \sum_{i \in N^{2t}(i_0)} w_t \cdot \eta_{\varphi(i)} \beta_i \\
&\qquad\quad = \sum_{t=1}^{T} \sum_{v \in \tau \cap V_{2T-2t}} w_t \cdot \eta_v = \sum_{\ell=0}^{T-1} \sum_{v \in \tau \cap V_{2\ell}} \omega_\ell \cdot \eta_v \\
&\qquad\qquad = \mathrm{val}_\omega(\phi(\beta); \eta). \quad \square
\end{aligned}$$

Let us define

$$\Pi_{p, d_L, d_R}(T, \omega) \overset{\text{def}}{=} \mathbb{P}_p \left\{ \min_{\tau} \mathrm{val}_\omega(\tau; \eta) \leqslant 0 \right\}\,,$$

where val is defined as in Definition 4. With this notation, Lemma 7 together with Theorem 4 (Local optimality implies LP optimality) has the following consequence.

LEMMA 8. *Let $p \in (0,1)$, G be a (d_L, d_R)-regular bipartite graph, $x \in \mathcal{C}(G)$ be a codeword, and $w \in [0,1]^T$ be a weight vector with $T < \frac{1}{4}\mathrm{girth}(G)$. Suppose y is obtained from x by flipping every bit independently with probability p. Then, codeword x is (T, w)-locally optimal with probability at least*

$$1 - n \cdot \Pi_{p, d_L, d_R}(T, \omega)\,, \quad \textit{where } \omega_\ell = w_{T-\ell}\,.$$

And with at least the same probability, x is also the unique optimal LP solution given y.

By virtue of Lemma 8, to understand the probability of LP decoding success, it is sufficient to estimate the probability $\Pi_{p, d_L, d_R}(T, \omega)$, for a given weight vector ω, bit error rate $p \in (0,1)$, and degrees (d_L, d_R). We give such estimates in the following subsection.

7.1 Bounding Processes on Trees by Evolving Laplace Transforms

We are going to study the probability of the existence of a negative value skinny subgraph in the (T, ω)-process in a recursive fashion, starting from the leaves of the tree $\mathcal{T}$.

We define the following correlated random variables Z_u for the vertices u of $\mathcal{T}$: The variable Z_u is equal to the minimum value of a skinny tree in the subtree $\mathcal{T}_u$

below the (variable or check) node u,

$$Z_u \overset{\text{def}}{=} \min_{\tau \cap \mathcal{T}_u \neq \emptyset} \sum_{\ell=0}^{T-1} \sum_{v \in \tau \cap V_{2\ell} \cap \mathcal{T}_u} \omega_\ell \cdot \eta_v \,.$$

Here, τ ranges over all skinny subtrees of $\mathcal{T}$.

Let $N^+(u)$ denote the set of neighbors of u that can be reached by one of its outgoing edges. The variables Z_u satisfy the following recurrence relations:

$$\begin{aligned} Z_{v_0} &= \sum_{v \in N^+(v_0)} Z_v \\ Z_u &= \omega_\ell \eta_u + \sum_{v \in N^+(u)} Z_v \qquad (u \in V_{2\ell},\ 0 \leqslant \ell < T) \\ Z_u &= \min_{v \in N^+(u)} Z_v \qquad (u \in V_{2\ell+1},\ 0 \leqslant \ell < T) \end{aligned}$$

Note that Z_{v_0} is just the minimum value of a skinny tree in the tree $\mathcal{T}$. Hence, $\Pi_{p,d_L,d_R}(T,\omega) = \mathbb{P}\{Z_{v_0} \leqslant 0\}$.

By symmetry, the distribution of a variable Z_u depends only on the height of vertex u. Also, for a fixed ℓ, the variables in $\{Z_u\}_{u \in V_\ell}$ are mutually independent, because for any two vertices u, u' of the same height ℓ, the subtrees $\mathcal{T}_u$ and $\mathcal{T}_{u'}$ are disjoint.

It follows that we can define random variables $X_0, \ldots, X_{T-1}, Y_0, \ldots, Y_{T-1}$ in the following way, so that X_ℓ has the same distribution as Z_u for $u \in V_{2\ell+1}$ and Y_ℓ has the same distribution as Z_u for $u \in V_{2\ell}$,

$$\begin{aligned} Y_0 &= \omega_0 \eta \\ X_\ell &= \min\left\{Y_\ell^{(1)}, \ldots, Y_\ell^{(d_R-1)}\right\} \qquad (0 \leqslant \ell < T) \\ Y_\ell &= \omega_\ell \eta + X_{\ell-1}^{(1)} + \ldots + X_{\ell-1}^{(d_L-1)} \qquad (0 < \ell < T) \end{aligned}$$

Here, η is a random variable that takes value 1 with probability $1-p$ and value -1 with probability p. The notation $X^{(1)}, \ldots, X^{(d)}$ means that we take d mutually independent copies of the random variable X (the copies are also independent of η in the last equation).

We use the Laplace transform of X_{T-1} in order to bound the probability $\Pi_{p,d_L,d_R}(T,\omega)$. Notice that in the following lemma and the remaining of this section, variable t is real-valued and has nothing to do with indexing levels of the tree; for that purpose we will be using instead the variables ℓ and s.

Lemma 9. *For every $t \geqslant 0$, $T \geqslant 1$,*

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

Proof. As noted before, $\Pi_{p,d_L,d_R}(T,\omega) = \mathbb{P}\{Z_{v_0} \leqslant 0\}$. Hence, by Markov's inequality

$$\Pi_{p,d_L,d_R}(T,\omega) = \mathbb{P}\left\{e^{-tZ_{v_0}} \geqslant 1\right\} \leqslant \mathbb{E}\, e^{-tZ_{v_0}} \,.$$

The variable Z_{v_0} is equal to the sum of the Z-values of its d_L children. Each child of the root v_0 has height $2T-1$ and hence its Z-value has the same distribution as

X_{T-1}. Using this and independence, we have as desired

$$\mathbb{E}\, e^{-tZ_{v_0}} = \left(\mathbb{E}\, e^{-tX_{T-1}}\right)^{d_L} . \quad \square$$

The following is our key lemma for estimating the probability $\Pi_{p,d_L,d_R}(T,\omega)$ (or more precisely, the Laplace transform of X_{T-1}). For the sake of brevity, let us denote $d_L' = d_L - 1$ and $d_R' = d_R - 1$.

Lemma 10. *For ℓ, s with $0 \leqslant s \leqslant \ell < T$, we have*

$$\mathbb{E}\, e^{-tX_\ell} \qquad \leqslant \qquad \left(\mathbb{E}\, e^{-tX_s}\right)^{{d_L'}^{\ell-s}} \qquad \cdot \qquad \prod_{k=0}^{\ell-s-1} \left(d_R'\, \mathbb{E}\, e^{-t\omega_{\ell-k}\eta}\right)^{{d_L'}^{k}} .$$

Proof. We derive the relation for $s = \ell - 1$. The general case follows by induction on $\ell - s$.

Since Y_ℓ is a sum of mutually independent variables,

$$\mathbb{E}\, e^{-tY_\ell} = \left(\mathbb{E}\, e^{-t\omega_\ell\eta}\right)\left(\mathbb{E}\, e^{-tX_{\ell-1}}\right)^{d_L'} .$$

We use a relatively crude estimate to bound the Laplace transform of X_ℓ in terms of the Laplace transform of Y_ℓ. By the definition of X_ℓ, we have $\exp(-tX_\ell) \leqslant \exp(-tY_\ell^{(1)}) + \ldots + \exp(-tY_\ell^{(d_R-1)})$ with probability 1. Hence,

$$\mathbb{E}\, e^{-tX_\ell} \leqslant d_R'\, \mathbb{E}\, e^{-tY_\ell} = \left(d_R'\, \mathbb{E}\, e^{-t\omega_\ell\eta}\right)\left(\mathbb{E}\, e^{-tX_{\ell-1}}\right)^{d_L'} ,$$

which is the desired bound for $s = \ell - 1$. $\square$

Armed with these general bounds on $\Pi_{p,d_L,d_R}(T,\omega)$ and the Laplace transform of X_ℓ, we can now derive several concrete bounds on $\Pi_{p,d_L,d_R}(T,\omega)$.

Uniform Weights. In this paragraph, we will consider the case $\omega = \mathbb{1}$. We apply Lemma 10 for $s = 0$. For brevity, let us denote $c_1 = \mathbb{E}\, e^{-tX_0}$ and $c_2 = d_R'\, \mathbb{E}\, e^{-t\eta}$. Note that $c_1 \leqslant c_2$ (using the same argument as in the proof of Lemma 10). For reasons that become apparent shortly, let us choose $t \geqslant 0$ so as to minimize $c := c_1 \cdot c_2^{1/(d_L-2)}$. We will assume $c < 1$. Now, the bound of Lemma 10 simplifies to

$$\begin{aligned} \mathbb{E}\, e^{-tX_\ell} \leqslant c_1^{{d_L'}^{\ell}} \cdot c_2^{\sum_{k=0}^{\ell-1} {d_L'}^{k}} &= {c_1}^{{d_L'}^{\ell}} \cdot \left(c_2^{1/(d_L'-1)}\right)^{{d_L'}^{\ell}-1} \\ &= c^{{d_L'}^{\ell}} \cdot c_2^{-1/(d_L'-1)} \leqslant c^{{d_L'}^{\ell}-1} . \end{aligned}$$

(To obtain the last inequality we used that $c_1 < 1$, which is implied by our assumption that $c < 1$, given that $c_1 \leqslant c_2$ and that $c = c_1 \cdot c_2^{1/(d_L-2)}$.) By Lemma 9 we can conclude from this bound that

$$\Pi_{p,d_L,d_R}(T,\mathbb{1}) \leqslant c^{d_L {d_L'}^{T-1} - d_L} .$$

Next, let us compute c_1 and c_2 as functions of p, d_L and d_R. The variable X_0 has the following distribution

$$X_0 = \begin{cases} +1, & \text{with probability } (1-p)^{d_R-1}, \\ -1, & \text{with probability } 1-(1-p)^{d_R-1}. \end{cases}$$

Hence,

$$c_1 = \mathbb{E}\, e^{-tX_0} = (1-p)^{d_R-1}\ e^{-t} + \left(1-(1-p)^{d_R-1}\right)\ e^{t}\,.$$

We also have

$$c_2 = (d_R - 1)\left((1-p)e^{-t} + pe^{t}\right)\,.$$

Putting together the calculations in this paragraph, we proved the following general bound on $\Pi_{p,d_L,d_R}(T,\mathbb{1})$.

LEMMA 11. *If $p \in (0,1)$ and $d_L, d_R > 2$ satisfy the condition*

$$c = \min_{t \geqslant 0}\Big\{\left((1-p)^{d_R-1}\ e^{-t} + \left(1-(1-p)^{d_R-1}\right)\ e^{t}\right) \cdot \left((d_R-1)\left((1-p)e^{-t}+pe^{t}\right)\right)^{1/(d_L-2)}\Big\} < 1\,,$$

then for $T \in \mathbb{N}_+$ and $\omega = (1,\dots,1) \in [0,1]^T$, we have

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant c^{d_L {d'_L}^{T-1} - d_L}\,.$$

For $(3,6)$-regular graphs, we have the following corollary.

COROLLARY 1. *Let $p \leqslant 0.02$, $d_L = 3$, and $d_R = 6$. Then, there exists a constant $c < 1$ such that for all $T \geqslant 1$ and $\omega = \mathbb{1}$,*

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant c^{2^{T-1}-1}\,.$$

Non-uniform Weights. In this paragraph, we will show how to improve the bounds by using different weights according to the height. We will use very simple weights: variable nodes at height 0 are weighted by a factor $\omega_0 \geqslant 0$, all variable nodes at higher heights are weighted by 1.

We apply Lemma 10 again for $s = 0$. As in the previous paragraph, the bound simplifies to

$$\mathbb{E}\, e^{-tX_\ell} \leqslant c^{{d'_L}^{\ell}} \cdot c_2^{-1/(d'_L-1)}\,,$$

where $c_1 = \mathbb{E}\, e^{-tX_0}$, $c_2 = d'_R\, \mathbb{E}\, e^{-t\eta}$, and $c = c_1 \cdot c_2^{1/(d_L-2)}$. The additional freedom of choosing the weight ω_0, allows us to minimize both c_1 and c_2 at the same time. To minimize c_2, we choose $t = \frac{1}{2}\ln\frac{1-p}{p}$. The value of c_1 is equal to

$$c_1 = \mathbb{E}\, e^{-tX_0} = (1-p)^{d_R-1}\ e^{-t\omega_0} + \left(1-(1-p)^{d_R-1}\right)\ e^{t\omega_0}\,,$$

which is minimized for

$$t\omega_0 = \ln\sqrt{\frac{(1-p)^{d_R-1}}{1-(1-p)^{d_R-1}}}\,.$$

Here, the right-hand side is nonnegative for $p < 1 - 2^{-1/d'_R}$. (Note that we do not have to worry whether $\omega_0 \leqslant 1$. By the definition of the (T,ω)-process, $\Pi_{p,d_L,d_R}(T,\omega)$ is invariant under (nonnegative) scaling of the weights.)

For these choices of ω_0 and t, we have

$$c_1 = 2\sqrt{(1-p)^{d_R-1}\left(1-(1-p)^{d_R-1}\right)}$$
$$c_2 = d'_R 2\sqrt{p(1-p)}\,.$$

Thus,

$$c = 2\sqrt{(1-p)^{d'_R}\left(1-(1-p)^{d'_R}\right)}\left(d'_R 2\sqrt{p(1-p)}\right)^{1/(d_L-2)}\,.$$

We proved the following bound on $\Pi_{p,d_L,d_R}(T,\omega)$ for $\omega = (\omega_0, 1, \ldots, 1)$.

Lemma 12. *If $p \in (0,1)$ and $d_L, d_R \geqslant 2$ satisfy the condition $p < 1 - 2^{-\frac{1}{d_R-1}}$ and $c_{p,d_L,d_R} < 1$, where*

$$c_{p,d_L,d_R} \overset{\text{def}}{=} 2\sqrt{(1-p)^{d_R-1}\left(1-(1-p)^{d_R-1}\right)}\Big((d_R-1)2\sqrt{p(1-p)}\Big)^{1/(d_L-2)}\,,$$

then there exists a constant $\omega_0 \geqslant 0$ such that for all $T \in \mathbb{N}_+$ and $\omega = (\omega_0, 1, \ldots, 1)$,

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant c' c^{d_L {d'_L}^{T-1}}\,,$$

where $c' = ((d_R-1)2\sqrt{p(1-p)})^{-d_L/(d_L-2)}$ and $c = c_{p,d_L,d_R}$.

Corollary 2. *Let $p \leqslant 0.0247$, $d_L = 3$, and $d_R = 6$. Then, there exist constants c' and $c < 1$ such that for all T,*

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant c' \cdot c^{2^{T-1}}\,, \qquad \textit{for some } \omega \in [0,1]^T\,.$$

7.2 Improved Bounds for $(3,6)$-Regular Trees

In this section we show how to obtain the bound of 0.05 on the tolerable error rate of $(3,6)$-regular codes. To achieve this we shall employ specially tailored non-uniform weights.

Let us first consider the weight vector $\bar{\omega} = (1, 2, \ldots, 2^s)$. Note that this weight vector has the effect that every level contributes equally to the ω-weighted value $\mathrm{val}_{\bar{\omega}}(\tau;\eta)$ of a skinny subtree τ. For a fixed value of s (say $s = 15$), we can compute the distribution of X_s explicitly using the recursive definition of the X and Y variables, since we deal with finite probability distributions. Hence, for a fixed s, we can also compute the value

$$\lambda_s \overset{\text{def}}{=} \min_{t \geqslant 0} \mathbb{E}\, e^{-tX_s}\,.$$

Let $t^* \geqslant 0$ be the point where the Laplace transform of X_s achieves its minimum λ_s. We now show how to bound $\Pi_{p,3,6}(T,\omega)$ in terms of λ_s for $\omega = (\bar{\omega}, \rho, \ldots, \rho) \in \mathbb{R}_+^T$, where ρ is a carefully chosen constant.

By Lemma 10, we have

$$\begin{aligned}\mathbb{E}\, e^{-t^* X_{T-1}} &\leqslant (\lambda_s)^{2^{T-s-1}} \left(5\,\mathbb{E}\, e^{-t^*\rho\eta}\right)^{\sum_{k=0}^{T-s-2} 2^k} \\ &= (\lambda_s)^{2^{T-s-1}} \left(5\,\mathbb{E}\, e^{-t^*\rho\eta}\right)^{2^{T-s-1}-1} \\ &= \left(\lambda_s \cdot 10\sqrt{p(1-p)}\right)^{2^{T-s-1}} \left(10\sqrt{p(1-p)}\right)^{-1},\end{aligned}$$

where we chose ρ such that $e^{t^*\rho} = \sqrt{(1-p)/p}$ so as to minimize $\mathbb{E}\, e^{-t^*\rho\eta}$. Using Lemma 9, we see that $\Pi_{p,3,6}(T,\omega)$ decreases doubly-exponentially in T if $\lambda_s \cdot 10\sqrt{p(1-p)} < 1$ for some s. We verified that this condition is satisfied for $s = 15$ and $p = 0.05$ using the numerical analysis software MATLAB. Hence, we establish the following theorem.

Theorem 6. *Let $p \leqslant 0.05$, $d_L = 3$, and $d_R = 6$. Then, there exists a constant $c < 1$ such that for $T > 15$,*

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant O\left(c^{2^{T-16}}\right) \qquad \textit{for some } \omega \in [0,1]^T .$$

We note that, as with all computer assisted proofs, it is possible that rounding errors or software and hardware bugs invalidate the numerical results.

We extend our approach to general (d_L, d_R)-regular trees in the following lemma.

Lemma 13. *Let $p \in (0, \frac{1}{2})$ and $d_L, d_R > 2$. Suppose that for some $s \in \mathbb{N}_+$ and some weight vector $\bar{\omega} \in \mathbb{R}^s_+$,*

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

Then, there exist constants $c < 1$, c' and $\rho \geqslant 0$ such that for all $T > s$,

$$\Pi_{p,d_L,d_R}(T,\omega) \leqslant c' \cdot c^{(d_L-1)^{T-s-1}} ,$$

where $\omega = (\bar{\omega}, \rho, \ldots, \rho) \in \mathbb{R}^T_+$.

8. CONCLUSIONS

One of our original intentions was to connect Belief Propagation to Linear Programming (or some other form of convex programming) and this remains open. It is unclear where to start since BP relies on highly nonlinear operations.

It would also be interesting to investigate if stronger versions of LP decoding using either lift-and-project operators such as Sherali Adams or using SDPs could have better provable performance for LDPCs, possibly approaching the information theoretic bound.

REFERENCES

Candes, E. and Tao, T. 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on 52,* 12 (dec.), 5406 –5425.

Chen, J., Dholakia, A., Eleftheriou, E., Fossorier, M., and Hu, X.-Y. 2005. Reduced-complexity decoding of ldpc codes. *Communications, IEEE Transactions on 53,* 8 (aug.), 1288 – 1299.

CHEN, J. AND FOSSORIER, M. 2002a. Density evolution for two improved bp-based decoding algorithms of ldpc codes. *Communications Letters, IEEE 6,* 5 (may), 208 –210.

CHEN, J. AND FOSSORIER, M. P. C. 2002b. Near optimum universal belief propagation based decoding of low-density parity check codes. *Communications, IEEE Transactions on 50,* 3, 406–414.

DASKALAKIS, C., DIMAKIS, A., KARP, R., AND WAINWRIGHT, M. 2008. Probabilistic analysis of linear programming decoding. *Information Theory, IEEE Transactions on 54,* 8 (aug.), 3565 –3578.

FELDMAN, J. AND KARGER, D. R. 2002. Decoding turbo-like codes via linear programming. In *Proc. of the 43rd annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 251–260.

FELDMAN, J., MALKIN, T., SERVEDIO, R. A., STEIN, C., AND WAINWRIGHT, M. J. 2007. Lp decoding corrects a constant fraction of errors. *Information Theory, IEEE Transactions on 53,* 1 (jan.), 82 –89.

FELDMAN, J., WAINWRIGHT, M., AND KARGER, D. 2005. Using linear programming to decode binary linear codes. *Information Theory, IEEE Transactions on 51,* 3 (march), 954 – 972.

FREY, B. J. AND KOETTER, R. 2001. The attenuated max-product algorithm. In *Advanced mean field methods*, M. Opper and D. Saad, Eds. MIT Press, Cambridge, MA, 213–227.

GALLAGER, R. G. 1963. *Low-density parity check codes.* MIT Press, Cambridge,MA.

GURUSWAMI, V., LEE, J. R., AND RAZBOROV, A. A. 2010. Almost euclidean subspaces of l_1^n via expander codes. *Combinatorica 30,* 1, 47–68.

INDYK, P. 2008. Explicit constructions for compressed sensing of sparse signals. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 30–33.

KOETTER, R. AND VONTOBEL, P. 2006. On the block error probability of lp decoding of ldpc codes. In *Inaugural Workshop of the Center for Information Theory and its Applications*. 6–10.

RICHARDSON, T. AND URBANKE, R. 2001. The capacity of low-density parity-check codes under message-passing decoding. *Information Theory, IEEE Transactions on 47,* 2 (feb), 599 –618.

SHERALI, H. D. AND ADAMS, W. P. 1990. A hierarchy of relaxation between the continuous and convex hull representations. *SIAM J. Discret. Math. 3*, 411–430.

SHOKROLLAHI, A. 2004. Ldpc codes: An introduction. In *Coding, cryptography and combinatorics*, K. Feng, H. Niederreiter, and C. Xing, Eds. Birkhauser, 85–110.

SIPSER, M. AND SPIELMAN, D. 1996. Expander codes. *Information Theory, IEEE Transactions on 42,* 6 (nov), 1710 –1722.

WIBERG, N. 1996. Codes and decoding on general graphs. Ph.D. thesis, Linkoping University, Sweden.