Reweighted LP Decoding for LDPC Codes

Amin Khajehnejad* Alexandros G. Dimakis† Babak Hassibi*
Benjamin Vigoda+ William Bradley+
*California Institute of Technology
†University of Southern California
+Lyric Semiconductors Inc. *

March 16, 2011

Abstract

We introduce a novel algorithm for decoding binary linear codes by linear programming. We build on the LP decoding algorithm of Feldman *et al.* and introduce a post-processing step that solves a second linear program that reweights the objective function based on the outcome of the original LP decoder output. Our analysis shows that for some LDPC ensembles we can improve the provable threshold guarantees compared to standard LP decoding. We also show significant empirical performance gains for the reweighted LP decoding algorithm with very small additional computational complexity.

1 Introduction

Linear programming (LP) decoding for binary linear codes was introduced by Feldman, Karger and Wainwright [2]. The method is based on solving a linear-programming relaxation of the integer program corresponding to the maximum likelihood (ML) decoding problem. LP decoding is connected to message-passing decoding [3, 4], and graph covers [5, 6] and has received substantial recent attention (see e.g. [6], and [7]).

As with the work described here, a related line of work has studied various improvements to either standard iterative decoding [8, 9] or to LP decoding via nonlinear extensions [10] or loop corrections [11].

The practical performance of LP decoding is roughly comparable to min- sum decoding and slightly inferior to sum-product decoding. In contrast to message-passing decoding, however, the

*This work was supported in part by the National Science Foundation under grants CCF-0729203, CNS-0932428 and CCF-1018927, by the Office of Naval Research under the MURI grant N00014-08-1-0747, by Caltech's Lee Center for Advanced Networking and by DARPA FA8750-07-C-0231.

LP decoder either concedes failure on a problem, or returns a codeword along with a guarantee that it is the ML codeword, thereby eliminating any undetected decoding errors.

The main idea of this paper is to add a second LP as a post-processing step when original LP decoding fails and outputs a fractional pseudocodeword. We use the difference between the input channel likelihood and the pseudocodeword coordinate to find a measure of disagreement or unreliability for each bit. We subsequently use this unreliability to bias the objective function and re-run the LP with the reweighted objective function. The reweighting increases the cost of changing reliable bits and decreases the cost for unreliable bits. We present an analysis that the provable BSC recovery thresholds improve for certain families of LDPC codes. We stress that the actual thresholds, even for the original LP decoding algorithm, remain unknown. Our analysis only establishes that the obtainable lower bounds on the fraction of recoverable errors are improved compared to the corresponding bounds for LP decoding. It is possible, however, that this is just an artifact of the lower bound techniques and that the true threshold is identical for both algorithms. In any case, the empirical performance gains we observe in our preliminary experimental analysis seem quite substantial.

A central idea in our analysis is a notion of robustness to changes in the BSC bit-flipping probability. This concept was inspired by a similar reweighted iterative ℓ_1 minimization idea for compressive sensing [21, 20]. We note that the reweighting idea of this paper involves changing the objective function of the LP from the reweighted max- product algorithm [12].

2 Basic Definitions

A vector $\mathbf{x}$ in $\mathbb{R}^n$ is called k-sparse if it has exactly k nonzero entries. The support set of a sparse vector $\mathbf{x}$ is the index set of its nonzero entries. If $\mathbf{x}$ is not sparse, the k-support set of $\mathbf{x}$ is defined as the index set of the maximum k entries of $\mathbf{x}$ in magnitude. We use $\|\mathbf{x}\|_p$ to denote the ℓ_p norm of a vector $\mathbf{x}$ for $p \geq 0$. in particular $\|\mathbf{x}\|_0$ is defined to be the number of nonzero entries in $\mathbf{x}$. For a set S, cardinality of S is denoted by $|S|$ and if $S \subset \{1, 2, \cdots, n\}$, then $\mathbf{x}_S$ is the sub-vector formed by those entries of $\mathbf{x}$ indexed in S. Also the complement set of S is denoted by S^c. The rate of a linear binary code $\mathcal{C}$ is denoted by R, and the corresponding parity check matrix is $H \in \mathbb{F}^{m \times n}$, where n is the length of each codeword and $m = Rn$. The factor graph corresponding to $\mathcal{C}$ is denoted by $\mathcal{G} = (X_v, X_c, \mathcal{E})$, where X_v and X_c are the sets of variable nodes and check nodes respectively, and $\mathcal{E}$ is the set of edges. For regular graphs, d_v and d_c denote the degree of variable and check nodes respectively. The girth of a graph $\mathcal{G}$, denoted by girth($\mathcal{G}$), is defined to be the size of the smallest cycle in $\mathcal{G}$.

3 Background

Suppose that C is a memoryless channel with binary input and an output alphabet $\mathcal{Y}$, defined by the transition probabilities $P_{Y|X}(y|x)$. For a received symbol y, the likelihood ratio is defined as $\log(\frac{P_{Y|X}(y|x=0)}{P_{Y|X}(y|x=1)})$, where x is the transmitted symbol. If a codeword $\mathbf{x}^{(c)}$ of length n from the linear code $\mathcal{C}$ is transmitted through the channel, and an output vector $\mathbf{x}^{(r)}$ is received, a maximum likelihood decoder can be used to estimate the transmitted codeword by finding the *most likely* transmitted input codeword. Let γ_i be the likelihood ratio assigned to the i^{th} received bit $\mathbf{x}_i^{(r)}$, and γ be the likelihood vector $\gamma = (\gamma_1, \cdots, \gamma_n)^T$. The ML decoder can be formalized as follows [1]

$$\begin{aligned} \text{ML decoder:} \quad & \text{minimize } \gamma^T \mathbf{x} \\ & \text{subject to } \mathbf{x} \in \text{conv}(\mathcal{C}), \end{aligned} \tag{1}$$

where $\text{conv}(\mathcal{C})$ is the convex hull of all the codewords of $\mathcal{C}$ in $\mathbb{R}^n$. The linear program (1) solves the ML decoding problem by the virtue of the fact that the objective $\gamma^T \mathbf{x}$ is minimized by a corner point (or vertex) of $\text{conv}(\mathcal{C})$, which is necessarily a codeword (In fact, vertices of $\text{conv}(\mathcal{C})$ are all the codewords of $\mathcal{C}$). In a linear program, the polytope over which the optimization is performed is described by linear inequalities describing the facets of the polytope. Since decoding for general linear codes is NP hard, it is unlikely that $\text{Conv}(\mathcal{C})$ can be efficiently described. Feldman *et al.* introduced a relaxation of (1) by replacing the polytope $\text{conv}(\mathcal{C})$ with a new polytope $\mathcal{P}$ that has much fewer facets, contains $\text{conv}(\mathcal{C})$ and retains the codewords of $\mathcal{C}$ as its vertices [1]. One way to construct $\mathcal{P}$ is the following. If the parity check matrix of $\mathcal{C}$ is the $m \times n$ matrix H and if $\mathbf{h}_j^T$ is the j-th row of H, then

$$\mathcal{P} = \cap_{1 \leq j \leq m} \text{conv}(\mathcal{C}_j), \tag{2}$$

where $\mathcal{C}_j = \{\mathbf{x} \in \mathbb{F}^n \mid \mathbf{h}_j^T \mathbf{x} = 0 \mod 2\}$. As mentioned earlier, with this construction, all codewords of $\mathcal{C}$ are vertices of $\mathcal{P}$. However, $\mathcal{P}$ has some additional vertices with fractional entries in $[0, 1]^n$. A vertex of the polytope $\mathcal{P}$ is called a *pseudo-codeword*. Moreover, if a pseudo-codeword is integral, i.e., if it has 0 or 1 entries, then it is definitely a codeword. The LP relaxation of (1) can thus be written as:

$$\begin{aligned} \text{LP decoder:} \quad & \text{minimize } \gamma^T \mathbf{x} \\ & \text{subject to } \mathbf{x} \in \mathcal{P}. \end{aligned} \tag{3}$$

The number of facets of $\mathcal{P}$ is exponential in the maximum weight of a row of H. Therefore, for LDPC codes with a small (often constant) row density, $\mathcal{P}$ has a polynomial number of facets, and it is possible to solve (3) in polynomial time.

For binary symmetric channels, (3) has another useful interpretation. In this case, rather than minimize $\gamma^T \mathbf{x}$ it turns out that one can alternatively minimize the Hamming distance between the output of the channel $\mathbf{x}^{(r)}$ and the individual codewords $\mathbf{x} \in \mathcal{C}$. Using the fact that the LP relaxation with $\mathcal{P}$ relaxes the entries of $\mathbf{x}$ from $x_i \in \{0, 1\}$ to $x_i \in [0, 1]$, we may replace the Hamming distance with the ℓ_1 distance $\|\mathbf{x} - \mathbf{x}^{(r)}\|_1$. This implies that the decoder (3) is equivalent to

$$\begin{aligned} \text{BSC-LP decoder:} \quad & \text{minimize } \|\mathbf{x} - \mathbf{x}^{(r)}\|_1 \\ & \text{subject to } \mathbf{x} \in \mathcal{P}. \end{aligned} \tag{4}$$

The above formulation can be interpreted as follows. For a received output binary vector $\mathbf{x}^{(r)}$, the solution to the LP decoder is basically the closest (in the ℓ_1 distance sense) pseudo-codeword to $\mathbf{x}^{(r)}$.

Linear programming decoding was first introduced by Feldman *et al.* [1, 2]. Subsequently [13] it was shown that if the parity check matrix is chosen to be the adjacency matrix of a high-quality expander, LP decoding can correct a constant fraction of errors. A fundamental lemma in [2] and used in the results therein, is that the LP polytope $\mathcal{P}$ is the same polytope from the view point of every codeword, and therefore for the analysis of LP decoding, it can be assumed without loss of generality that the transmitted codeword is the all zero codeword. The theoretical results of [13] were based on a dual witness argument, i.e. a feasible set of variables that set the dual of LP equal to zero. However, the bounds on success threshold of LP decoding achieved by this technique is considerably smaller than the empirical recovery threshold of LP decoder in practice. A later analysis of LP decoding by Daskalakis *et al.* [14] improved upon those bounds for random expander codes, through employing a different dual witness argument, and considering a *weak* notion of LP success rather than the *strong* notion of [13]. A strong threshold means that *every* set of errors of up to a certain size can be corrected, whereas a weak threshold implies that *almost all* error sets of a certain size are recoverable. Note that there is a gap of about one order of magnitude between the error-correcting thresholds of [14] and the ones observed in practice.

The arguments of [13] and [14] are based on the existence of dual certificates that guarantee the success of the LP decoder and require codes that are based on bipartite expander graphs. A more recent work of Arora *et al.* uses a quite different certificate based on the primal LP problem

[15]. This approach results in fairly easier computations and significantly better thresholds for LP decoding. However, the underlying codes discussed in [15] are based on factor graphs with a large girth (at least doubly logarithmic in the number of variables), rather than unbalanced expanders considered in previous arguments. Note that similar to [14], the bounds of [15] are weak bounds, certifying that for a random set of errors up to a fraction of bits, LP decoding succeeds with high probability. The largest such fraction is called the weak recovery threshold.

A somewhat related problem to the LP decoding of linear codes is the compressed sensing (CS) problem. In CS an unknown real vector $\mathbf{x}$ of size n is to be recovered from a set of m linear measurement, represented by $\mathbf{y} = A\mathbf{x}$, where $A \in \mathbb{R}^{m \times n}$, and $m << n$. This is in general infeasible, since the measurement matrix A is under-determined and the resulting system of equations is ill-posed, i.e., it can have infinitely many solutions. However, imposing a sparsity condition on $\mathbf{x}$ can make the solution unique. The unique sparse solution can be found by exhaustive search for instance, which is formulated by the following minimization program:

$$\begin{aligned} &\text{minimize } \|\mathbf{x}\|_0 \\ &\text{subject to } A\mathbf{x} = \mathbf{y}. \end{aligned} \tag{5}$$

Since (5) is NP-hard, one possible approximation is relaxing the ℓ_0 norm of x to the closet convex norm $\|\mathbf{x}\|_1$, which results in the following ℓ_1 minimization program:

$$\text{minimize } \|\mathbf{x}\|_1 \tag{6}$$

$$\text{subject to } A\mathbf{x} = \mathbf{y}. \tag{7}$$

(7) is a linear program, which can in general be solved in polynomial time. There has been substantial theoretical work on this linear programming relaxation, see *e.g.* [18, 19, 23, 24, 26]

Recently, systematic connections between the problems of channel coding LP and CS ℓ_1 relaxation has been found [16, 17]. In this paper, we build on those connections to improve LP decoding, and further extend the ideas of robustness and reweighted ℓ_1 minimization in compressed sensing to channel coding LP.

4 Extended Certificate and Robustness of LP decoder

The success of LP decoder is often certified by the existence of a *dual witness* [13, 14]. Similarly, for ℓ_1 minimization in the context of CS, a dual witness certificate can guarantee that the recovery of sparse signals is successful [22]. However, it has proven more promising to express the success

condition of ℓ_1 minimization in terms of the properties of the null space of the measurement matrix [23, 24, 25]. The condition is called *null space property*, through which it is possible to characterize one class of "good" measurement matrices for CS, namely matrices that are congruent with ℓ_1 minimization decoding. The advantage of the null space interpretation, apart from the fact that it results in sharper analytical bounds, is that with proper parametrization, it can also be used to evaluate the performance of ℓ_1 minimization in the presence of noise. This is known as the *robustness* of ℓ_1 minimization. A consequence of the robustness property is that when ℓ_1 minimization fails to recover a sparse signal, it often gives a decent approximation to it [20]. To the best of our knowledge, a similar certificate has not been introduced in the context of channel coding linear programming. In other words, when LP decoding fails to return an integral solution, it is not known how far in the proximity of the actual codeword it lies. We provide an approximate solution to this question in this section, using the following strategy. We introduce a property called fundamental cone property for an arbitrary code $\mathcal{C}$, and show that for binary symmetric channels, this is related to the robustness of the solution of the LP decoder. The robustness of LP decoding has two consequences. First, it implies that the linear program is tolerant to a limited mismatch in the available formulation. Second, it can be used to develop iterative schemes that improve the performance of the decoder. We will discuss these issues in proceeding sections. We begin by defining the fundamental cone of a code from [16].

Definition 1. *Let H be a parity check matrix. Define $\mathcal{J}$ and $\mathcal{I}$ to be the set of rows and columns of H. Also, for each $j \in \mathcal{J}$, define $\mathcal{I}_j = \{i \in \mathcal{I} \mid H(j,i) = 0\}$. The fundamental cone, $\mathcal{K}(H)$, of H is the set of all vectors $\omega = (\omega_1, \omega_2, \ldots, \omega_n)^T$ that satisfy*

$$\omega_i \geq 0, \qquad \forall 1 \leq i \leq n, \tag{8}$$

$$\omega_i \leq \sum_{i' \in \mathcal{I}_j \setminus i} \omega_{i'}, \quad \forall j \in \mathcal{J} \ \forall i \in \mathcal{I}_j. \tag{9}$$

$\mathcal{K}(H)$ is the smallest cone in $\mathbb{R}^n$ that encompasses the polytope $\mathcal{P}$. If a vector lies on an edge of $\mathcal{K}$, it is called a *minimal pseudo-codeword*. For simplicity, in the sequel, we use $\mathcal{K}$ instead of $\mathcal{K}(H)$ whenever there is no ambiguity.

Definition 2. *Let $S \subset \{1, 2, \cdots, n\}$ and $C \geq 1$ be fixed. A code $\mathcal{C}$ with parity check matrix H is said to have the fundamental cone property FCP(S, C), if for every nonzero vector $\omega \in \mathcal{K}(H)$ the following holds:*

$$C\|\omega_S\|_1 < \|\omega_{S^c}\|_1, \tag{10}$$

if for every index set S of size k, $\mathcal{C}$ has the FCP(S, C), then we say that $\mathcal{C}$ has the fundamental cone property FCP(k, C).

In the next lemma we show how the fundamental cone property can be used to evaluate the performance of an LP decoder, even when it fails to recover the true codeword. The key assumption is that the channel is a bit flipping channel (*e.g.* BSC).

Lemma 4.1. *Let $\mathcal{C}$ be a code that has the FCP(S, C) for some index set S and some $C \geq 1$. Suppose that a codeword $\mathbf{x}^{(c)}$ from $\mathcal{C}$ is transmitted through a bit flipping channel, and the received codeword is $\mathbf{x}^{(r)}$. If the pseudocodeword $\mathbf{x}^{(p)}$ is the output of LP decoder for the received codeword $\mathbf{x}^{(r)}$, then the following holds:*

$$\|\mathbf{x}^{(p)} - \mathbf{x}^{(c)}\|_1 < 2\frac{C+1}{C-1}\|(\mathbf{x}^{(r)} - \mathbf{x}^{(c)})_{S^c}\|_1. \tag{11}$$

Proof. Without loss of generality, we may assume that the all zero codeword was transmitted, i.e. $\mathbf{x}^{(c)} = 0$. We have

$$\begin{aligned}
\|\mathbf{x}_S^{(r)}\|_1 + \|\mathbf{x}_{S^c}^{(r)}\|_1 &= \|\mathbf{x}^{(r)}\|_1 \\
&\overset{(a)}{\geq} \|\mathbf{x}^{(p)} - \mathbf{x}^{(r)}\|_1 \\
&= \|(\mathbf{x}^{(p)} - \mathbf{x}^{(r)})_S\|_1 + \|(\mathbf{x}^{(p)} - \mathbf{x}^{(r)})_{S^c}\|_1 \\
&\overset{(b)}{\geq} \|\mathbf{x}_S^{(r)}\|_1 - \|\mathbf{x}_S^{(p)}\|_1 + \|\mathbf{x}_{S^c}^{(p)}\|_1 - \|\mathbf{x}_{S^c}^{(r)}\|_1.
\end{aligned} \tag{12}$$

(a) is true because from (4), $\|\mathbf{x}^{(p)} - \mathbf{x}^{(r)}\|_1 \leq \|\mathbf{x}^{(c)} - \mathbf{x}^{(r)}\|_1$. Also (b) holds by the triangular inequality. Note that $\mathbf{x}^{(p)} \in \mathcal{K}(H)$, so by definition, $C\|\mathbf{x}_S^{(p)}\|_1 < \|\mathbf{x}_{S^c}^{(p)}\|_1$. This implies that

$$\|\mathbf{x}_{S^c}^{(p)}\|_1 - \|\mathbf{x}_S^{(p)}\|_1 > \frac{C-1}{C+1}\|\mathbf{x}^{(p)}\|_1. \tag{13}$$

Applying this to the left hand side of (12) we obtain

$$2\frac{C+1}{C-1}\|\mathbf{x}_{S^c}^{(r)}\|_1 > \|\mathbf{x}^{(p)}\|_1, \tag{14}$$

Which is the desired result.

■

An asymptotic case of Lemma 4.1 for $C \to 1$ is in fact equivalent to the LP success condition. Namely, let S be the index set of the flipped bits in the transmitted codeword, i.e. the set of bits that differ in $\mathbf{x}^{(r)}$ and $\mathbf{x}^{(c)}$. If FCP(S, C) holds for some $C > 1$, then Lemma 4.1 implies that LP decoding can successfully recover the original codeword. Now let us say that the set of

errors (flipped bits) is slightly larger than S, and does include S. Then the vector $(\mathbf{x}^{(r)} - \mathbf{x}^{(c)})_{S^c}$ has a few (but not too many) nonzero entries. Therefore, even if the LP decoder output $\mathbf{x}^{(p)}$ is not equal to the actual codeword, it is still possible to obtain an upper bound on its ℓ_1 distance to the unknown codeword. We recognize this as the robustness of LP decoder, and characterize it by FCP(S, C), for $C > 1$. Furthermore, two notions of robustness can be considered. Strong robustness means that for *every* set S of up to some cardinality k, the FCP condition holds, namely FCP(k, S). Weak robustness on the other hand deals with almost all sets S of up to a certain size. In the next section we present a thorough analysis of LP robustness for two categories of codes: expander codes and codes with $\Omega(\log \log n)$ girth. For these two classes of codes, rigorous analysis has been done on the performance of LP decoders in [13, 14] and [15], respectively. We build on the existing arguments to incorporate the robustness condition and analyze the fundamental cone property. Afterwards, we discuss the implications of LP robustness.

5 Analysis of LP Robustness

In most cases, if there exists a certificate for the success of LP decoder, it can be often extended to guarantee that the LP decoder is robust, namely that the FCP condition is satisfied for some $C > 1$. By carefully re-examining the analysis of LP decoder, one might be able to do such a generalization. This is the main focus of this section. We consider three major methods that exist in the literature for analyzing the performance of LP decoders. The first one is due to Feldman et. al [13], and is based on using a dual witness type of argument to certify the success of LP decoder for expander graphs. The second one is that of Daskalakis *et al.* [14], which again considers linear programming decoding in expander codes. Specifically, [14] analyzes the dual of LP and finds a simple combinatorial condition for the dual value to be zero (implying that the LP decoder is successful). The condition is basically the existence of a so-called *hyperflow* from the set of flipped bits to unflipped bits. The existence of a valid hyperflow can be secured by the presence of so-called (p, q)-matchings. It then follows from a detailed series of probabilistic calculations that (p, q)-matchings of interest exist for certain expander codes. The main difference between this analysis and that of Feldman *et al.* is the probabilistic nature of the arguments in [14], which account for weak recovery thresholds.

A third analysis of the LP decoder was done by Arora *et al.*, [15], which is based on factor graphs with a doubly logarithmic girth. Unlike previous dual feasibility arguments, the authors in [15] introduce a certificate in the primal domain, which is of the following form: If in the primal LP problem, the value of the objective function for the original codeword is smaller than its value for all vectors within a local deviation from the original codeword, then LP decoder

succeeds. Local deviations are defined by weighted minimal local trees whose induced subgraphs are cycle-free.

5.1 Strong LP Robustness for Expander Codes

Strong thresholds of LP decoding for expander codes are derived in [13]. To show that the transmitted codeword is the LP optimal obtained by (3) when a subset of the bits are flipped, a set of feasible dual variables are found that satisfy the following conditions. Suppose the factor graph of $\mathcal{C}$ is denoted by $\mathcal{G} = (X_v, X_c, \mathcal{E})$. We may also assume without loss of generality that the all zero codeword was transmitted. A set of feasible dual variables is defined as follows (see [13] for more details)

Definition 3. *For an error set S, a set of feasible dual variables is a labeling of the edges of the factor graph $\mathcal{G}$, say $\{\tau_{ij} \mid v_i \in X_v \; c_j \in X_c\}$, where the following two conditions are satisfied:*

i) For every check node $c_j \in X_c$ and every two disjoint neighbors of c_j, say $v_i, v_{i'} \in N(j)$, we have $\tau_{ij} + \tau_{i'j} \geq 0$.

ii) For every variable node $v_i \in X_v$, we have $\sum_{c_j \in N(v_i)} \tau_{ij} \leq \gamma_i$.

We show that a generalized set of dual feasible variables can be used to derive LP robustness. To this end, we show that the existence of a set of feasible dual variables implies the FCP condition. The following lemma is proved in Appendix A.

Lemma 5.1. *Suppose that a set of dual variables satisfy the feasibility conditions (Definition 3) for an arbitrary log-likelihood vector γ. Then for every vector $\omega \in \mathcal{K}(\mathcal{C})$, the following holds*

$$\sum_{1 \leq i \leq n} \gamma_i \omega_i > 0. \tag{15}$$

A special case of Lemma 5.1 is when the channel is a BSC, and a set S of the bits have been flipped. We can also assume without loss of generality that the all zero codeword was transmitted. Then Lemmas 4.1 and 5.1 imply that if a dual feasible set exists, then LP decoder succeeds, which is the conclusion of [13]. In this case the log-likelihood vector γ takes the value -1 over the set S and 1 over the set S^c. Let us now define a new likelihood vector γ' by

$$\gamma' = \begin{cases} -C & i \in S \\ 1 & i \in S^c \end{cases}, \tag{16}$$

for some $C > 1$. If a dual feasible set exists that satisfies the feasibility condition for γ', then it follows that FCP(S, C) holds. Knowing this and pursuing an argument very similar to [13] for

the construction of dual feasible in expander codes, we are able to prove the following lemma, the proof of which is given in Appendix B.

Theorem 5.1. *Let $\mathcal{G}$ be the factor graph of a code $\mathcal{C}$ of length n and rate $R = \frac{m}{n}$, and let $\delta > 2/3 + 1/d_v$. If $\mathcal{G}$ is a bipartite $(\alpha n, \delta d_v)$ expander graph, then $\mathcal{C}$ has FCP(t, C), where $t = \frac{3\delta-2}{2\delta-1}\alpha$ and $C = \frac{2\delta-1}{2\delta-1-1/d_v}$. This means that for every every set S of size t, FCP(t, C) holds.*

Basically, [13] shows that if the conditions of Theorem 5.1 are satisfied, then LP succeeds for every error set of size t, namely that FCP$(t, 1)$ holds. However Theorem 5.1 asserts that, in addition, a strong robustness holds, i.e. FCP(t, C) for some $C > 1$.

5.2 Weak LP Robustness for Expander Codes

We show that for random expander codes a probabilistic analysis similar to the dual witness analysis of [14] can be used to find the extents of the fundamental cone property for expander codes, in a weak sense. We rely on the matching arguments of [14], with appropriate adjustments. The following definition is given in [14].

Definition 4. *For nonnegative integers p and q, and a set F of variable nodes, a (p, q)-matching on F is defined by the following conditions:*

(a) each bit $v_i \in F$ must be matched with p distinct check nodes, and

(b) each variable node $v_{i'} \in F^c$ must be connected with

$$X_{i'} := \max\{q - d_v + Z_{i'}, 0\} \tag{17}$$

checks nodes from the set $N(F)$, that are different from the check nodes that the nodes in F are matched to, where $Z_{i'}$ is defined as $Z_{i'} := |N(i') \cap N(F)|$.

We prove the following lemma that relates the existence of a (p, q)-matching to the fundamental cone property of a code $\mathcal{C}$. This lemma is proved in Appendix C.

Lemma 5.2. *Let $\mathcal{C}$ be a code of rate R with a bipartite factor graph $\mathcal{G}$, where every variable node has degree d_v. Let S be a subset of the variable nodes of $\mathcal{G}$. If a (p, q)-matching on S exists, then $\mathcal{C}$ has the FCP$(S, \frac{2p-d_v}{d_v-q})$.*

[14] provides a probabilistic tool for the existence of (p, q)-matchings in regular bipartite expander graphs, which helps answer the question of how large an error set LP decoding can fix. For example, for a random LDPC(8,16) code, the probabilistic analysis implies that with high probability, a fraction 0.002 of errors is recoverable using LP decoder. However, taking the

specifications of the matching that leads to this conclusion and applying Lemma 5.2, it turns out that for an error set of size $0.002n$, the robustness factor is at least $C = 1.3$, i.e the code has FCP$(0.002n, 1.3)$.

5.3 Weak LP Robustness for Codes with $\Omega(\log\log(n))$ Girth

Recall that $\mathcal{G} = (X_v, X_c, \mathcal{E})$ is used to denote the factor graph of the parity check matrix H (or of code $\mathcal{C}$), where X_v and X_c are the sets of variable and check nodes respectively and $\mathcal{E}$ is the set of edges. Also recall that the girth of $\mathcal{G}$ is defined as the size of the shortest cycle in $\mathcal{G}$. Without loss of generality, we assume that $X_v = \{v_1, v_2, \cdots, v_n\}$, where v_i is the variable node corresponding to the i^{th} bit of the codeword. Let $T \leq \frac{1}{4}\text{girth}(\mathcal{G})$ be fixed. The following notions are defined in [15].

Definition 5. *A tree $\mathcal{T}$ of height $2T$ is called a skinny subtree of $\mathcal{G}$, if it is rooted at some variable node v_{i_0}, for every variable node v in $\mathcal{T}$ all the neighboring check nodes of v in $\mathcal{G}$ are also present in $\mathcal{T}$, and for every check node c in $\mathcal{T}$ exactly two neighboring variable nodes of c in $\mathcal{G}$ are present in $\mathcal{T}$.*

Definition 6. *Let $\mathbf{w} \in [0,1]^T$ be a fixed vector. A vector $\beta^{(\mathbf{w})}$ is called a minimal T-local deviation, if there is a skinny subtree of $\mathcal{G}$ of height $2T$, say $\mathcal{T}$, so that for every variable node v_i $1 \leq i \leq n$,*

$$\beta_i^{(\mathbf{w})} = \begin{cases} \mathbf{w}_{h(i)} & \text{if } v_i \in \mathcal{T} \setminus \{v_{i_0}\} \\ 0 & \text{otherwise} \end{cases},$$

where $h_i = \frac{1}{2}d(v_{i_0}, v_i)$.

The key to the derivations of [15] is the following lemma:

Lemma 5.3 (Lemma 1 of [15])**.** *For any vector $\mathbf{z} \in \mathcal{P}$, and any positive vector $\mathbf{w} \in [0,1]^T$, there exists a distribution on the minimal T-local deviations $\beta^{(\mathbf{w})}$, such that*

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

where $0 < \alpha \leq 1$.

Lemma 5.3 has the following interpretation. If a linear property holds for all minimal T-local deviations (*e.g.* $f(\beta^{(\mathbf{w})}) \geq 0$, where $f(.)$ is a linear operator), then it also holds for all pseudo-codewords (i.e. $f(\mathbf{z}) \geq 0\ \forall \mathbf{z} \in \mathcal{P}$). Interestingly enough, the robustness of LP decoding for a given set of bit flips S has a linear certificate, namely FCP(S, C)[1]. In other words, if we define:

[1]Note that this is only true for bit fliping channels, where the output alphabet in the binary field.

$$f_C^{(S)}(\mathbf{x}) = \sum_{i \in S^c} x_i - C \sum_{i \in S} x_i,$$

then FCP(S, C) holds, if and only if $f_1^{(S)}(\mathbf{z}) \geq 0$ for every pseudocodeword $\mathbf{z} \in \mathcal{P}$. Therefore, according to Lemma 5.3, it suffices that the condition be true for all T-local deviations. Furthermore, for arbitrary $C > 1$, if $f_C^{(S)}(\beta^{(\mathbf{w})}) \geq 0$ for all minimal T-local deviations $\beta^{(\mathbf{w})}$, then it follows that the code has the FCP(S, C) property. This simple observation helps us extend the probabilistic analysis of [15] to robustness results for LP decoding. The resulting key theorem is mentioned below, the proof of which can be found in Appendix D. In order to state the theorem, first we define η to be a random variable that takes the value $-C$ with probability p and value 1 with probability $1 - p$. Also, define the sequences of random variables $X_i, Y_i,\ i \geq 0$, in the following way:

$$\begin{aligned} Y_0 &= \eta, \\ X_i &= \min\{Y_i^{(1)}, \ldots, Y_i^{(d_c-1)}\} \quad \forall i > 0, \\ Y_i &= 2^i \eta + X_{i-1}^{(1)} + \cdots + X_{i-1}^{(d_v-1)} \quad \forall i > 0, \end{aligned} \tag{18}$$

Where $X^{(j)}$s are independent copies of a random variable X.

Theorem 5.2. *Let $0 \leq p \leq 1/2$ be the probability of bit flip, and S be the random set of flipped bits. If for some $j \in \mathbb{N}$,*

$$c = \gamma^{1/(d_v-2)} \min_{t \geq 0} \mathbb{E} e^{-tX_j} < 1,$$

where $\gamma = (d_c-1)\frac{C_R+1}{C_R}(\frac{C_R \cdot p}{1-p})^{1/(C_R+1)}(1-p) < 1$, Then with probability at least $1 - O(n)c^{d_v(d_v-1)^{T-1}}$ the code $\mathcal{C}$ has the FCP(S, C), where T is any integer with $j \leq T < 1/4 girth(\mathcal{G})$.

For $d_c = 6$ and $d_v = 3$, a lower bound on the robustness parameter C that results from Theorem 5.2 is plotted against the probability of bit flip p, in Figure 1.

6 Implications of LP robustness

6.1 Mismatch Tolerance

One of the direct consequences of the robustness of LP decoding is that if there is a slight mismatch in the implementation of the LP decoder, its performance does not degrade significantly.

More formally, suppose that due to noise, quantization or some other factor, a mismatched log-likelihood vector $\gamma' = \gamma + \Delta\gamma$ is used in the LP implementation. We refer to such a decoder as a *mismatched LP decoder*. Since the channel is BSC, the entries of γ all have the same amplitude g. We also define $\delta = \max_i |\Delta\gamma_i|$, and assume that $\delta < g$. We can prove the following theorem.

Theorem 6.1. *Suppose that S is the set of bit errors. Let $C = \frac{g+\delta}{g-\delta}$. If $\mathcal{C}$ has FCP(S, C), then the mismatched LP decoder corrects all errors and recovers the original codeword.*

Proof. We assume without loss of generality that the all zero codeword is transmitted. We show that if FCP(S, C) holds, then the all zero codeword is the minimum cost vector in the polytope $\mathcal{P}$. Suppose ω is a nonzero vector in the fundamental code $\mathcal{K}$. We begin with the definition of FCP(S, C) and write

$$-C\sum_{i\in S}\omega_i + \sum_{i\in S^c}\omega_i > 0. \tag{19}$$

Multiply both sides by $(g - \delta)$:

$$-\sum_{i\in S}(g+\delta)\omega_i + \sum_{i\in S^c}(g-\delta)\omega_i > 0. \tag{20}$$

We also know from the definition of δ that $\gamma'_i > (g - \delta)$ for $i \in S^c$, and $\gamma'_i > -g - \delta$ for $i \in S^c$, and that $\omega \geq 0$. Therefore

$$-\sum_{i\in S\cup S^c}\gamma'_i\omega_i > 0, \tag{21}$$

which proves that the all zero codeword is the unique minimum cost solution of the mismatched LP.

■

6.2 Pseudocodewords and High Error Rate Subsets

We showed in Section 4 that for an appropriate code $\mathcal{C}$, even when LP decoder fails to recover an actual codeword from the output of a BSC, the ℓ_1 distance between the obtained pseudocodeword and the actual codeword can be bounded by a finite factor of excess errors (see equation 11). We now show that this property allows us to use the output of LP decoder to find a *high error* rate subset of the bits of linear size, namely a subset of bits over which the fraction of errors is significantly larger than the fraction of errors in the entire received codeword. Obtaining such *importance* subset is very crucial, since it provides additional information about a significant

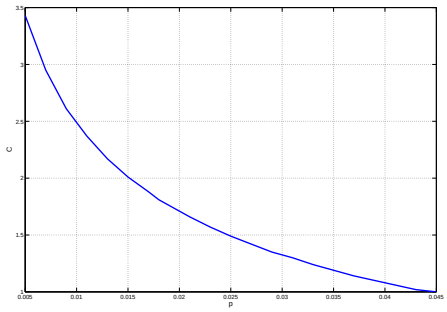

Figure 1: Approximate upper bound for the robustness factor C as a function of error probability p for $d_c = 6$ and $d_v = 3$, based on Theorem 5.2.

proportion of the bits which can be used to improve the decoder's performance. For instance, one can impose additional soft or hard constraints on the importance subset, and solve a constrained linear program or other post processing algorithms following the initial linear program. This forms the idea for the proposed iterative LP decoding algorithm which will be outlined in Section 7.

Consider a code $\mathcal{C}$ of length n and rate R, and a codeword $\mathbf{x}^{(c)}$ from $\mathcal{C}$ transmitted through a bit flipping channel. Suppose that a set K of the bits get flipped, where the cardinality of K is $(1+p^*)\epsilon n$ for some $0 < p^* < 1$ and $\epsilon > 0$. Denote the received vector by $\mathbf{x}^{(r)}$. We are interested in the case where LP fails, so the LP minimal $\mathbf{x}^{(p)}$ is a fractional pseudocodeword. However, the size of the error set is only slightly larger than the correctable size p^*n. In other words, we assume that for some subset $K_1 \subset K$ of size p^*n, the code has FCP(K_1, C), for some $C > 1$. We show in the next lemma that the index set of the largest k entries of the vector $\mathbf{x}^{(r)} - \mathbf{x}^{(p)}$ has a significant overlap with K with high probability, and is thus a high error rate subset of entries. The following theorem formalized this claim.

Theorem 6.2. *Suppose that a codeword $\mathbf{x}^{(c)}$ is transmitted through a bit flipping channel, and the output $\mathbf{x}^{(r)}$ differs from the input in a set K of the bits with $|K| = p^*(1+\epsilon)n$, for some $0 < p^* < 1$ and $\epsilon > 0$. Also, suppose that for a subset $K_1 \subset K$ of size p^*n, FCP(K_1, C) holds, for some $C > 1$, and that the LP minimal is the pseudocodeword $\mathbf{x}^{(p)}$. If L is the set of the $p^*(1+\epsilon)n$ largest entries of the vector $\mathbf{x}^{(r)} - \mathbf{x}^{(p)}$ in magnitude, then the fraction of errors in $\mathbf{x}^{(r)}$ over the set L is at least $1 - 2\frac{C+1}{C-1}\epsilon$.*

Before proving this theorem, we state the following definition and lemma.

Definition 7. *Let $\mathbf{x} \in \mathbb{R}^n$ be a k-sparse vector. For $\lambda > 0$, We define $W(x, \lambda)$ to be the size of the largest subset of nonzero entries of $\mathbf{x}$ that has a ℓ_1 norm less than or equal to λ, i.e.,*

$$W(\mathbf{x}, \lambda) := \max\{|S| \mid S \subseteq supp(\mathbf{x}),\ \|\mathbf{x}_S\|_1 \leq \lambda\}. \tag{22}$$

The following Lemma is proven in [20].

Lemma 6.1 (Lemma 1 of [20]). *Let* $\mathbf{x}$ *be a* k*-sparse vector and* $\hat{\mathbf{x}}$ *be another vector. Also, let* K *be the support set of* $\mathbf{x}$ *and* L *be the* k*-support set of* $\hat{\mathbf{x}}$*, namely the set of* k *largest entries of* $\hat{\mathbf{x}}$*. If* $d = \|\mathbf{x} - \hat{\mathbf{x}}\|_1$*, then*

$$|K \cap L| \geq k - W(\mathbf{x}, d). \tag{23}$$

Proof of Theorem 6.2. Define $k = p^*(1+\epsilon)n$, and apply Lemma 6.1 to the k-sparse vector $\mathbf{x}^{(r)} - \mathbf{x}^{(c)}$, and the vector $\mathbf{x}^{(p)} - \mathbf{x}^{(r)}$. If L is the index set of the largest k entries of $\mathbf{x}^{(p)} - \mathbf{x}^{(r)}$ in magnitude, then from Lemma 6.1 we have

$$|K \cap L| \geq k - W(\mathbf{x}^{(r)} - \mathbf{x}^{(c)}, \Delta), \tag{24}$$

where $\Delta = \|\mathbf{x}^{(c)} - \mathbf{x}^{(p)}\|_1$. Since $\|\mathbf{x}^{(r)} - \mathbf{x}^{(c)}\|$ has only ± 1 nonzero entries, (24) can be written as

$$|K \cap L| \geq k - \|\mathbf{x}^{(c)} - \mathbf{x}^{(p)}\|_1. \tag{25}$$

We use the inequality in (11) to further lower bound the right hand side of (25). Recall that $K_1 \subset K$ is such that $\mathcal{C}$ has FCP(K_1, C). Therefore, we can write:

$$\begin{aligned} |K \cap L| &\geq k - 2\frac{C+1}{C-1}\|(\mathbf{x}^{(r)} - \mathbf{x}^{(c)})_{K_1^c}\|_1 & (26)\\ &= k - 2\frac{C+1}{C-1}(k - p^*n). & (27) \end{aligned}$$

Dividing both sides by $|K| = k$, we conclude that at least a fraction $1 - 2\frac{C+1}{C-1}\epsilon$ of the set L are flipped bits. ■

7 Iterative Reweighted LP Algorithm and Improved Strong Threshold

First, we briefly define different recovery thresholds for LP decoding for more clarity of the statements that will follow. In general, the actual weak and strong thresholds for a given classes of linear codes might be unknown, and the existing threshold only provide lower bounds on these quantities. For expander codes for instance, the size of the error set that can be recovered via

LP can be lower bounded by the size of the set for which a dual witness exists [13, 14]. Since a dual witness is only a sufficient condition for the success of LP decoding, the actual thresholds are generally expected to be higher. However, to date, the best achievable thresholds for LP decoding for expander codes are those given by the dual feasibility arguments. Therefore, we also consider thresholds associated with those limits, namely the "provable" thresholds. Specifically, we define the following four thresholds for LP decoding on a given code $\mathcal{C}$ that has regular variable and check degrees d_v and d_c.

Definition 8 (Recovery thresholds)**.** *Strong recovery threshold is denoted by p_s^*, and is defined as the largest fraction such that* every *set of size $p_s^* n$ is recoverable via LP decoding. Weak recovery thresholds is denoted by p_w^*, and it means that* almost all *sets of size $p_w^* n$ is recoverable via LP. We define p_{sd}^* to be the maximum provable strong threshold achieved by a dual feasible, [13]. Similarly, p_{wd}^* is the provable weak threshold, i.e. for almost all sets of size $p_{wd}^* n$, a dual feasible ([14]) exist.*

As sketched in Theorem 6.2, by examining the deviation of the LP optimal (pseudo-codeword) and the received vector, it is possible to identify a high error rate (HER) subset of bits in which the fraction of bit flips is higher than the overall probability of error, or the fraction of errors in the complement of the HER set. One way this imbalancedness can be exploited is by using a weighted LP scheme. This is outlined in the following iterative algorithm.

Algorithm 1.

1. *Run LP decoding. If the output is integral terminate, otherwise proceed.*
2. *Take the fractional pseudocodeword $\mathbf{x}^{(p)}$ from the LP decoder, and construct the deviation vector $\mathbf{x}^{(d)} = \mathbf{x}^{(r)} - \mathbf{x}^{(p)}$.*
3. *Sort the entries of $\mathbf{x}^{(d)}$ in terms of absolute value, and denote by L the index set of its* ***smallest*** *pn entries.*
4. *solve the following weighted LP:*

$$\min_{\mathbf{x}\in\mathcal{P}} \lambda_1 \|(\mathbf{x}-\mathbf{y})_L\|_1 + \lambda_2 \|(\mathbf{x}-\mathbf{y})_{L^c}\|_1, \tag{28}$$

where λ_1 and λ_2, where $\lambda_1 < 0$ ad $\lambda_2 > 0$ are fixed parameters.

Algorithm 1 is only twice as complex as LP decoding. We prove in the following that algorithm 1 has a strictly improved provable strong and weak recovery thresholds than the dual feasibility thresholds p_{sd}^* and p_{wd}^* (Recall the definitions of p_{sd}^* and p_{sd}^* from Definition 8).

Theorem 7.1. *For any code $\mathcal{C}$, there exist $\epsilon_1 > 0$,$\epsilon_2 > 0$, $\lambda_1 < 0$ and $\lambda_2 > 0$ so that every error set of size $(1+\epsilon_1)p^*_{sd}$, and almost all error sets of size $(1+\epsilon_2)p^*_{wd}$ can be corrected by Algorithm 1.*

we start with the following lemma

Lemma 7.1. *Suppose a codewords $\mathbf{x}$ transmitted is through a binary channel. Also suppose that the bits of $\mathbf{x}$ can be divided into two sets L and L^c, so that at least a fraction p_1 of the bits in L are flipped, and at most a fraction p_2 of the bits in L^c are flipped. Then the following weighted LP decoding*

$$\min_{\mathbf{x}\in\mathcal{P}} -\|(\mathbf{x}-\mathbf{y})_L\|_1 + \|(\mathbf{x}-\mathbf{y})_{L^c}\|_1, \tag{29}$$

can recover $\mathbf{x}$, provided that

$$(1-p_1)|L| + p_2|L^c| \leq p^*_{sd}. \tag{30}$$

Proof. We assume without loss of generality that the all zero codeword has been transmitted and prove that there exists a feasible dual (Definition 3) for the LP decoder 29. The feasible dual must satisfy condition (i) of Definition 3 for all check nodes, and in addition:

$$\sum_{j\in N(i)} \tau_{ij} \leq \begin{cases} 1 & i \in L \cap S \\ -1 & i \in L \cap S^c \\ -1 & i \in L^c \cap S \\ 1 & i \in L^c \cap S^c \end{cases} . \tag{31}$$

One can note that the conditions of (31) are equivalent to τ_{ij}'s being a feasible dual set for ordinary LP decoder when the error set is $S_1 = (L \cap S^c) \cup (L^c \cap S)$. Therefore if the size of S_1 is smaller than $p^*_{sd}n$, from the definition of p^*_{sd}, such a feasible dual set exists. This completes the proof the theorem. ■

proof of Theorem 7.1. We set $\lambda_1 = -1$ and $\lambda_2 = 1$. Suppose the all zero codeword have been transmitted without loss of generality, and the received binary vector $x^{(r)}$ has pn errors, where $p = (1+\epsilon_0)p^*_{sd}$. From Theorem 5.1, $\mathcal{C}$ has FCP$(p^*_s n, C)$ for some $C > 1$. Therefore, if we apply Theorem 6.2 to the output of LP, namely $\mathbf{x}^{(p)}$, we conclude that the set L of most pn deviated bits in $\mathbf{x}^{(p)}$ with respect to $\mathbf{x}^{(r)}$, and the set S of the errors in $\mathbf{x}^{(r)}$, have at least a fraction $1 - 2\frac{C+1}{C-1}\epsilon_1$ overlap. Define $p_1 = \frac{|L\cap S|}{|L|}$ and $p_2 = \frac{|L^c\cap S|}{|L^c|}$. We must have

$$p_1 \geq 1 - 2\frac{C+1}{C-1}\epsilon_0, \tag{32}$$

$$p_1|L| + p_2|L^c| = p. \tag{33}$$

Therefore, as $\epsilon_0 \to 0$, $p_1 \to 1$ and $p_2 \to 0$. So, for some small enough ϵ_0, the following will eventually hold

$$(1-p_1)|L| + p_2|L^c| \leq p^*_{sd}. \tag{34}$$

Thus, according to Lemma 7.1, the weighted LP step of Algorithm 1 corrects all errors. similarly, if a random set of pn bits are flipped, when $p = (1+\epsilon_2)p^*_{wd}$, from Lemma 5.2 we conclude that with high probability there exists a $C > 1$ so that FCP(S_1, C) holds for a random subset S_1 of the bit errors of size $p^*_{wd}n$. Therefore, using Theorem 6.2, it follows that the set L of most pn deviated bits in $\mathbf{x}^{(p)}$ with respect to $\mathbf{x}^{(r)}$, and the set of errors in $\mathbf{x}^{(r)}$ have at least an overlap fraction of $1 - 2\frac{C+1}{C-1}\epsilon_2$. The remainder of the proof is the same as the previous case, i.e. by applying Lemma 7.1. ■

8 Simulations

We have implemented Algorithm 1 on a random LPDC code of size $n = 1000$ and rate $R = 3/4$ and have compared the results with other existing methods. The variable node degree is $d_v = 3$, and thus, $d_c = 4$. The algorithm is compared with the mixed integer method of Draper and Yedidia [27], and the random facet guessing algorithm of [28]. The mixed integer algorithm re-runs the LP decoding by setting integer constraints on a small subset of "least certain" bits, namely the positions where the LP minimal pseudocodeword entries are closest to 0.5. We have taken the size of the constrained subset to be $M = 5$, which means the number of extra iterations is 32 for the mixed integer method. We also choose to run 20 more extra random iterations for facet guessing. In random facet guessing, a face (facet) of the polytope $\mathcal{P}$ is selected at random, among all the faces on which the LP minimal pseudocodeword does not reside. Then, LP decoder is re-run with the additional constraint that the solution is on the selected face. In contrast, Algorithm 1 has only one extra iteration. All methods are simulated in MATLAB where LP decoder is implemented via the cvx toolbox [29]. We have plotted the BER curves versus the probability of error p in Figure 2. For Algorithm 1, for each p, we have experimentally found the optimal λ_1 and λ_2 by choosing the values that on average result in the best performance. For most of the cases the chosen values where in the ranges $-3 \leq \lambda_1 \leq -0.5$ and $1 \leq \lambda_2 \leq 3$. Observe the superior BER performance of Algorithm 1 which becomes more significant for smaller values of p. For $p = 0.11$, the BER improvement in the reweighted LP method is at least one order of magnitude. In our preliminary experimental evaluation we observe that the BER curves eventually collapse into the same curve as the LP curve, except for the reweighted LP algorithm, which is an indication of the fact that the empirical thresholds of

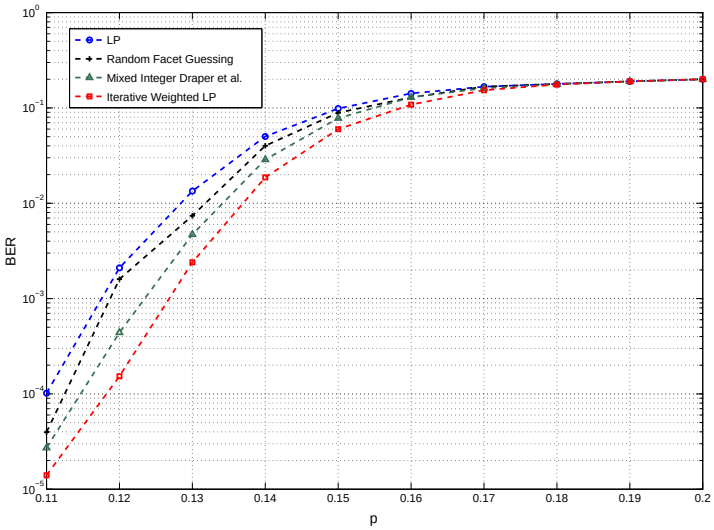

Figure 2: BER curves as a function of channel flip probability p, for LP decoding and different iterative schemes; random facet guessing of [28], mixed integer method of [27], and the suggested iterative reweighted LP of Algorithm 1. The code is a random LDPC(3,4) of length $n = 1000$.

Algorithm 1 are better than those of LP decoder and existing polynomial time post processing methods.

References

[1] J. Feldman, "Decoding Error-Correcting Codes via Linear Programming", PhD thesis, MIT, 2003.

[2] J. Feldman, M. J. Wainwright, and D. R. Karger,"Using linear programming to decode binary linear codes," IEEE Transactions on Information Theory, 51(3):954-972, 2005.

[3] J. Feldman, D. R. Karger and M. J. Wainwright, "Linear Programming-Based Decoding of Turbo-Like Codes and its Relation to Iterative Approaches,"Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing Oct. 2002.

[4] M. J. Wainwright, T. S. Jaakkola and A. S. Willsky,"MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches," Proc. Allerton Conference on Communication, Control and Computing Oct. 2002.

[5] R. Koetter and P. O. Vontobel, "Graph-covers and iterative decoding of finite length codes," Proc. 3rd International Symp. on Turbo Codes, Sep. 2003.

[6] P. O. Vontobel and R. Koetter,"Towards low-complexity linear-programming decoding," Proc. Int. Conf. on Turbo Codes and Related Topics,Munich, Germany, Apr. 2006.

[7] M. H. Taghavi and P. H. Siegel,"Adaptive linear programming decoding," IEEE Int. Symposium on Information Theory, Seattle, WA, July 2006.

[8] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," IEEE Transactions on Information Theory, May 2001, pp:908-917.

[9] H. Pishro-Nik and F. Fekri, "On Decoding of LDPC Codes over the Erasure Channel," IEEE Trans. Inform. Theory, Vol. 50, pp:439-454 2004.

[10] K. Yang, J. Feldman and X. Wang "Nonlinear programming approaches to decoding low-density parity-check codes," IEEE J. Sel. Areas in Communication, Vol. 24 NO. 8, pp: 1603-1613, Aug. 2006.

[11] M. Chertkov and V. Y. Chernyak,"Loop calculus helps to improve belief propagation and linear programming decoding of LDPC codes", Allerton Conference on Communications, Control and Computing,Monticello, IL, Sep. 2006.

[12] M. J. Wainwright and T. S. Jaakkola and A. S. Willsky,"Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing," IEEE Trans. Information Theory, Vol. 51, NO. 11 pp: 3697-3717, Nov. 2005.

[13] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, " LP decoding corrects a constant fraction of errors", In Proc. IEEE International Symposium on Information Theory, 2004.

[14] C. Daskalakis, A. G. Dimakis, R. M. Karp and M. J. Wainwright, "Probabilistic Analysis of Linear Programming decoding", IEEE Transactions on Information Theory, Volume 54, Issue 8, Aug. 2008.

[15] S. Arora, D. Steuer, and C. Daskalakis, "Message-Passing Algorithms and Improved LP Decoding", ACM STOC 2009.

[16] A.G. Dimakis and P. O. Vontobel, "LP Decoding meets LP Decoding: A Connection between Channel Coding and Compressed Sensing", Allerton 2009.

[17] A. G. Dimakis, R. Smarandache and P. O. Vontobel "LDPC Codes for Compressed Sensing", arxiv 1012.0602.

[18] D. Donoho, "High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension," Discrete and Computational Geometry, 102(27), pp. 617-652 2006, Springer.

[19] D. Donoho and J. Tanner, "Neighborlyness of randomly-projected simplices in high dimensions," Proc. National Academy of Sciences, 102(27), pp. 9452-9457, 2005.

[20] A. Khajehnejad, W. Xu, S. Avestimehr, B. Hassibi, "Improved Sparse Recovery Thresholds with Two-Step Reweighted ℓ_1 Minimization", ISIT 2010.

[21] E. J. Candés, M. B. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted l1 Minimization", *Journal of Fourier Analysis and Applications*, 14(5), pp. 877-905, special issue on sparsity, December 2008.

[22] E. J. Candès and T. Tao, "Decoding by linear programming", IEEE Trans. Inform. Theory, 51 4203-4215

[23] D.L. Donoho and X. Huo," Uncertainty principles and ideal atomic decomposition", IEEE Transactions on Information Theory, 47(7):28452862, 2001.

[24] M.Stojnic, W. Xu, and B. Hassibi, "Compressed sensing - probabilistic analysis of a null-space characterization" *IEEE International Conference on Acoustic, Speech and Signal Processing, ICASSP 2008*

[25] A. Cohen, W. Dahmen, and R. DeVore," Compressed sensing and best k-term approximation", Journal of the American Mathematical Society ,Volume 22, Number 1, January 2009, Pages 211231

[26] W. Xu and B. Hassibi, "On Sharp Performance Bounds for Robust Sparse Signal Recoveries", accepted to *the International Symposium on Information Theory 2009.*

[27] S.C. Draper, J.S. Yedidia, Y. Wang, "ML Decoding via Mixed-Integer Adaptive Linear Programming", IEEE International Symposium on Information Theory (ISIT), June 2007 (ISIT 2007, TR2007-022).

[28] A. G. Dimakis, A. A. Gohari and M. Wainwright, "Guessing Facets: Polytope Structure and Improved LP Decoder", IEEE Transactions on Information Theory, Volume 55, Issue 8, Aug. 2009,

[29] cvx toolbox webpage http://cvxr.com/cvx/.

A Proof of Lemma 5.1

We first prove the following lemma.

Lemma A.1. *Suppose* $\{\tau_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ *is a set of feasible dual variables on the edges of the factor graph* $\mathcal{G}$ *of the code* $\mathcal{C}$*, for some arbitrary log-likelihood vector* γ*. Then for every vector* $\mathbf{w} \in \mathcal{K}(\mathcal{C})$ *and every check node* c_j*, the following holds*

$$\sum_{v_i \in N(c_j)} w_i \tau_{ij} \geq 0. \tag{35}$$

Proof. We only use condition (i) of a feasible set of dual variables. Note that among the variable nodes in $N(c_j)$, there can be at most one node v_i with $\tau_{i,j} < 0$. Let v_i be such a variable node. From the definition of $\mathcal{K}$ we can write

$$w_i \leq \sum_{i' \in N(j) \backslash i} w_{i'},$$

or equivalently:

$$\tau_{ij} w_i + \sum_{v_{i'} \in N(v_j) \backslash i} |\tau_{ij}| w_{i'} \geq 0. \tag{36}$$

Moreover, we know that $\tau_{ij} + \tau_{i'j} \geq 0$ for $i' \neq i$, from the condition (i) of the dual feasibility. Therefore, replacing $\tau_{i'j}$ with $|\tau_{ij}|$ for each $i' \neq i$ does not decrease the left hand side of (36), and thus

$$\sum_{v_i \in N(c_j)} w_i \tau_{ij} \geq 0.$$

■

We now invoke Lemma A.1 that for every check node c_j, $\sum_{v_i \in N(c_j)} w_i \tau_{ij} \geq 0$. If we sum these inequalities for all check nodes c_j we obtain:

$$\sum_{c_j \in X_c} \sum_{v_i \in N(c_j)} w_i \tau_{ij} = \sum_{v_i \in X_v} w_i \sum_{c_j \in N(v_i)} \tau_{ij} \geq 0,$$

When X_v and X_c are the sets of variable and check nodes respectively. Since τ_{ij}s are feasible dual variables, from condition (ii) of feasibility (Definition 3), we must have $\sum_{c_j \in N(v_i)} \tau_{ij} < \gamma_i$. It then follows that

$$\sum_{v_i \in X_v} \gamma_i w_i > 0.$$

B Proof of Theorem 5.1

We basically repeat the argument of [13] with some slight adjustments. Let S be the set of flipped bits, or interchangeably the set of corresponding variable nodes in the factor graph $\mathcal{G}$ (we use v_i to refer to the variable node corresponding to the i^{th} bit).

Definition 9 ((δ, λ) matching from [13])**.** *A (δ, λ) matching of the set S is a set M of edges of the factor graph $\mathcal{G}$, so that no two edges are connected to the same check node, every node in S is connected to at least δd_v edges of M, and every node in S' is connected to at least λd_v edges of M. Here S' is the set of variable nodes that are connected to at least $(1-\lambda)$ check nodes in $N(S)$.*

If there is a (δ, λ) matching on the set S, then we consider the following labeling of the edges of $\mathcal{G}$. For a check node v_j, if it is adjacent to an edge τ_{ij} is M then set $\tau_{ij} = -x$ and $\tau_{i'j} = x$ for every other variable node $v_i' \in N(v_j)$ $i' \neq i$. Otherwise, label all of the edges of the edges adjacent to j by 0. It can be seen that this for this labeling $\{\tau_{ij}\}$ satisfies condition (i) of dual feasibility (Definition 3), and furthermore:

$$\sum_{j \in N(i)} \tau_{ij} \leq \begin{cases} (1-2\delta) d_v x & i \in S \\ (1-\lambda) d_v x & i \in S^c \end{cases}. \tag{37}$$

We know take $\lambda = 2 - 2\delta + 1/d_v$. Let us define a new likelihood vector γ' by

$$\gamma' = \begin{cases} -C & i \in S \\ 1 & i \in S^c \end{cases} . \tag{38}$$

If a dual feasible set exists that satisfies the feasibility condition for the vector γ', then this implies that the FCP(S, C) holds. Now, since $C < \frac{2\delta-1}{1-\lambda}$, if we choose x to be

$$x = \frac{1}{(1-\lambda)d_v}, \tag{39}$$

then, it is clear that $(1-2\delta)d_v x < -C$. So the dual feasibility condition is satisfied, if we can construct the required (δ, λ) matching for S. From [13], if $|S| \leq \frac{3\delta-2}{2\delta-1}\alpha$, and $\mathcal{G}$ is a bipartite $(\alpha n, \delta d_v)$ expander, the desired matching exists. This proves that FCP(S, C) holds. Since this argument holds for every set S of size $t = \frac{3\delta-2}{2\delta-1}\alpha$, we conclude that $\mathcal{C}$ has FCP(t, C).

C Proof of lemma 5.2

Consider a vector ω in the fundamental cone $\mathcal{K} = \mathcal{K}(H)$ of the parity check matrix H. Without loss of generality, we may assume that $S = \{1, 2, \cdots, t\}$. For each $1 \leq i \leq t$, let the neighbors of the variable node v_i in the (p, q)-matching on S be denoted by $c_1^i, c_2^i, \cdots, c_p^i$. The check nodes c_j^i are $p \times t$ distinct nodes. From the definition of $\mathcal{K}$, if $\omega \in \mathcal{K}$, then for each c_j^i we may write:

$$\omega_i \leq \sum_{l \in N(c_i^j) \setminus v_i} \omega_l, \quad \forall 1 \leq i \leq t \; 1 \leq j \leq p. \tag{40}$$

We add all inequalities of (40) for $1 \leq i \leq t$ and $1 \leq j \leq p$. For $i \leq t$, ω_i appears exactly p times on the left hand side of the sum and, at most $d_v - p$ times on the right. For $i > t$, ω_i appears in at most $d_v - q$ inequalities and on the right hand side. This comes directly from the definition of a (p, q)-matching on the set S. Therefore

$$p \sum_{i \in S} \omega_i \leq (d_v - p) \sum_{i \in S} \omega_i + (d_v - q) \sum_{i \in S^c} \omega_i, \tag{41}$$

and thus,

$$\frac{2p - d_v}{d_v - q} \sum_{i \in S} \omega_i \leq \sum_{i \in S^c} \omega_i, \tag{42}$$

which proves that $\mathcal{C}$ has the desired fundamental cone property.

D Proof of Theorem 5.2

We denote the set of variable nodes and check nodes by X_v and X_c respectively. For a fixed $\mathbf{w} \in [0,1]^T$, let $\mathcal{B}$ be the set of all minimal T-local deviations, and $\mathcal{B}_i$ be the set of minimal T-local deviations that result from a skinny tree rooted at the variable node v_i. Also, assume S is the random set of flipped bits, when the flip probability is p. Interchangeably, we also use S to refer to the set of variable nodes corresponding to the flipped bits indices. We are interested in the probability that for all $\beta^{(\mathbf{w})} \in \mathcal{B}$, $f_C^{(S)}(\beta^{(\mathbf{w})}) \geq 0$. Recall that

$$f_C^{(S)}(\mathbf{x}) := \sum_{i \in S^c} x_i - C \sum_{i \in S} x_i.$$

For simplicity we denote this event by $f_C^{(S)}(\mathcal{B}) \geq 0$. Since the bits are flipped independently and with the same probability, we have the following union bound

$$\mathbb{P}\left(f_C^{(S)}(\mathcal{B}) \geq 0\right) \geq 1 - n\mathbb{P}\left(f_C^{(S)}(\mathcal{B}_1) \geq 0\right). \tag{43}$$

Now consider the full tree of height 2T, that is rooted at the node v_1, and contains every node u in $\mathcal{G}$ that is no more than $2T$ distant from v, i.e. $d(v_1, u) \leq 2T$. We denote this tree by $B(v_1, 2T)$. To every variable node u of $B(v_1, 2T)$, we assign a label, $I(u)$, which is equal to $-C\omega_{h(u)}$ if $u \in S$, and is $\omega_{h(u)}$ if $u \in S^c$, where $(\omega_0, \omega_2, \cdots, \omega_{2T-2}) = \mathbf{w}$. We can now see that the event $f_C^{(S)}(\mathcal{B}_1) \geq 0$ is equivalent to the event that for all skinny subtrees $\mathcal{T}$ of $B(v_1, 2T)$ of height $2T$, the sum of the labels on the variable nodes of $\mathcal{T}$ is positive. In other words, if Γ_1 is the set of all skinny trees of height $2T$ that are rooted at v_1, then $f_C^{(S)}(\mathcal{B}_1) \geq 0$ is equivalent to:

$$\min_{\mathcal{T} \in \Gamma_1} \sum_{v \in \mathcal{T} \cap X_v} I(v) \geq 0. \tag{44}$$

We assign to each node u (either check or variable node) of $B(v_1, 2T)$ a random variable Z_u, which is equal to the contribution to the quantity $\min_{\mathcal{T} \in \Gamma_1} \sum_{v \in \mathcal{T} \cap X_v} I(v)$ by the offspring of the node u in the tree $B(v_1, 2T)$, and the node u itself. The value of Z_u for can be determined recursively from all of its children. Furthermore, the distribution of Z_u only depends on the height of u in $B(v_1, 2T)$. Therefore, to find the distribution of Z_u, we use $X_0, X_1, \cdots, X_{T-1}$ as random variables with the same distribution as Z_u when u is a variable node (X_0 is assigned to the lowest level variable node) and likewise $Y_1, Y_2, \cdots, Y_{T-1}$ for the check nodes. It then follows that:

$$
\begin{aligned}
Y_0 &= \omega_0 \eta, \\
X_i &= \min\{Y_i^{(1)}, \ldots, Y_i^{(d_c-1)}\} \quad \forall i > 0, \\
Y_i &= \omega_i \eta + X_{i-1}^{(1)} + \cdots + X_{i-1}^{(d_v-1)} \quad \forall i > 0,
\end{aligned}
\tag{45}
$$

where $X^{(j)}$s are independent copies of a random variable X, and η is a random variable that takes the value $-C$ with probability p and value 1 with probability $1-p$. It follows that

$$
\begin{aligned}
\mathbb{P}\left(f_C^{(S)}(\mathcal{B}_1) \leq 0\right) &= \mathbb{P}\left(X_{T-1}^{(1)} + \cdots + X_{T-1}^{(d_v)} \leq 0\right) \\
&\leq (\mathbb{E}(e^{-tX_{T-1}}))^{d_v}.
\end{aligned}
\tag{46}
$$

The last inequality is by Markov inequality and is true for all $t > 0$. The rest of the proof we bring here is basically appropriate modifications of the derivations of [15] for the Laplace transform evolution of the variables X_is and Y_is, to account for a non-unitary robustness factor C. By upper bounding the Laplace transform of the variables recursively it is possible to show that (see Lemma 8 of [15], the argument is completely the same for our case)

$$
\begin{aligned}
\mathbb{E}e^{-tX_i} \leq & \left(\mathbb{E}e^{-tX_j}\right)^{(d_v-1)^{i-j}} \\
& \prod_{0 \leq k \leq i-j-1} \left((d_c-1)\mathbb{E}e^{-t\omega_{i-k}\eta}\right)^{(d_v-1)^k},
\end{aligned}
\tag{47}
$$

for all $1 \leq j \leq i < T$.

If we take the weight vector as $\omega = (1, 2, \cdots, 2^j, \rho, \rho, \cdots, \rho)$ for some integer $1 \leq j < T$, and use equation (47), we obtain:

$$
\begin{aligned}
\mathbb{E}e^{-tX_{T-1}} \leq & \left(\mathbb{E}e^{-tX_j}\right)^{(d_v-1)^{T-j-1}} \\
& \cdot \left((d_c-1)\mathbb{E}e^{-t\rho\eta}\right)^{\frac{(dv-1)^{T-j-1}-1}{d_v-2}}.
\end{aligned}
$$

ρ and t can be chosen to jointly minimize $\mathbb{E}e^{-tX_j}$ and $\mathbb{E}e^{-t\rho\eta}$ in the above, which along with

(46) results in

$$\begin{aligned}\mathbb{P}(f_C^{\mathcal{S}}(\mathcal{B}_1 \leq 0)) &\leq (\mathbb{E}e^{-tX_{T-1}})^{d_v} \\ &\leq \gamma^{-d_v/(d_v-2)} \times c^{d_v(d_v-1)^{T-j-1}},\end{aligned}$$

where $\gamma = (d_c - 1)\frac{C+1}{C}(1-p)(\frac{C.p}{1-p})^{1/(C+1)}$ and $c = \gamma^{1/(d_v-2)} \min_{t\geq 0} \mathbb{E}e^{-tX_j}$. If $c < 1$, then probability of error tends to zero as stated in Theorem 5.2.