LP Decoding Corrects a Constant Fraction of Errors

(CORC Technical Report TR-2003-08)

Jon Feldman* — Columbia University
Tal Malkin† — Columbia University
Rocco A. Servedio‡ — Columbia University
Cliff Stein§ — Columbia University
Martin J. Wainwright¶ — UC Berkeley

Abstract

We show that for low-density parity-check (LDPC) codes with sufficient expansion, the *Linear Programming (LP) Decoder* of Feldman, Karger and Wainwright (Allerton, 2003) can correct a constant fraction of errors. More specifically, we show that if the Tanner graph modeling the code has regular left degree c and expands by a factor more than δc on all variable node sets of size at most αn, where $\delta > \frac{2}{3} + \frac{1}{3c}$, then the LP decoder succeeds as long as at most $\frac{3\delta-2}{2\delta-1}(\alpha n - 1)$ bits are flipped by the channel. A random regular graph will have sufficient expansion with high probability, and recent work by Capalbo *et al.* shows that such graphs can be constructed efficiently. A key element of our method is the construction of a zero-valued *dual* solution to the decoding linear program.

Our result implies that the word error rate of the LP decoder decreases exponentially in the code length under the binary symmetric channel (BSC). This is the first such result for LP decoding; the only previously known performance bounds were an inverse-polynomial word error rate bound for high-girth cycle codes, and a proof that at least $\Theta(n^{1-\epsilon})$ errors can be corrected in general LDPC codes under bit-flipping channels.

The results given here are stronger than all known finite-length results under the BSC for message-passing decoders such as min-sum and sum-product (belief propagation). Recent work by Koetter and Vontobel (Turbo Codes, 2003) shows that LP decoding and min-sum decoding of LDPC codes are closely related by the "graph cover" structure of their pseudocodewords; in their terminology, our result implies that the *pseudodistance* of these graph covers under the BSC can grow linearly in the length of the code.

*Department of Industrial Engineering and Operations Research, Columbia University, New York, NY. email: jonfeld@ieor.columbia.edu.

Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

†Department of Computer Science, Columbia University, New York, NY. email: tal@cs.columbia.edu.

‡Department of Computer Science, Columbia University, New York, NY. email: rocco@cs.columbia.edu.

§Department of Industrial Engineering and Operations Research, Columbia University, New York, NY. email: cliff@ieor.columbia.edu.

Research partially supported by NSF Grant DMI-9970063 and an Alfred P. Sloan Foundation Fellowship.

¶Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, CA. email: wainwrig@eecs.berkeley.edu.

Research partially supported by NSF grant IIS-9988642, ARO MURI DAA19-02-1-0383.

1 Introduction

Turbo codes [3] and low-density parity-check codes [14] have been the focus of intense study in the last ten years. While their observed error-correcting performance is unparalleled, the theoretical understanding of this behavior remains incomplete. One very successful technique for analyzing the average behavior of message-passing decoders on code ensembles with large block length is *density evolution* [21, 20, 18]. This technique, however, fails to explain the behavior of message-passing algorithms on specific finite-length codes when the messages traverse cycles in the underlying Tanner graph [24].

Successful finite-length analysis of a sub-optimal decoding algorithm for LDPC codes requires a useful combinatorial description of the output space, or the set of *pseudocodewords* associated with the decoder. Under the binary erasure channel, Di *et al.* [6] identified the message-passing pseudocodewords as "stopping sets," and used this characterization to give decoding performance results. For more general channels, Wiberg [25] (see also [12, 13]) used the *computation tree* of the message-passing decoder to analyze pseudocodewords; however, this analysis has not led to finite-length performance bounds, except for limited families of codes.

The *Linear Programming (LP) Decoder* of Feldman, Karger and Wainwright [9] provides an alternative to message-passing decoding that is more amenable to finite-length analysis. Specific LP decoders have been defined for turbo codes [10] and LDPC codes [11, 8]. The pseudocodewords for an LP decoder are the vertices of a linear polytope whose constraints depend on the structure of the code. In general, LP pseudocodewords unify many known notions of pseudocodewords for various codes and decoders (see [9]). For the case of LDPC codes, Koetter and Vontobel [16, 17] described these LP pseudocodewords as "graph covers," and established a connection to the pseudocodewords of the message-passing "min-sum" algorithm.

Until now, the only known performance results for the LP decoder used graphs with high girth; this yielded an inverse-polynomial word error rate (WER) bound for rate-1/2 repeat-accumulate codes [10, 7], and a proof that at least $\Theta(n^{1-\epsilon})$ errors can be corrected in general LDPC codes under bit-flipping channels. In this paper, we show that LP decoders can correct up to a constant fraction of error in bit-flipping channels. Using a simple Chernoff bound, this implies WER $\leq e^{-\Omega(n)}$ under the BSC. This is the first proof that LP decoding has an inverse-exponential word error rate on a constant rate code. Furthermore, no such word error rate bound is known for message-passing decoders such as min-sum and sum-product (belief propagation) on finite-length LDPC codes.

Our result is based on the *expansion* of the Tanner graph, rather than its girth. (Recall that a Tanner graph $\mathcal{G}$ is a (k, Δ)-*expander* if for all sets S of variable nodes where $|S| \leq k$, at least $\Delta|S|$ check nodes are incident to S.) With this definition, our main theorem is given explicitly as follows:

Theorem 1 *Let $\mathcal{C}$ be a low-density parity-check code with length n and rate at least $1 - m/n$ described by a Tanner graph $\mathcal{G}$ with n variable nodes, m check nodes, and regular left degree c. Suppose $\mathcal{G}$ is an $(\alpha n, \delta c)$-expander, where $\delta > 2/3 + 1/(3c)$ and δc is an integer. Then the LP decoder succeeds, as long at most $\frac{3\delta-2}{2\delta-1}(\alpha n - 1)$ bits are flipped by the channel.*

Random Tanner graphs will meet the conditions of this theorem with high probability, and recent work by Capalbo *et al.* [4] gives efficient deterministic constructions of such graphs.

The proof of Theorem 1 is based on showing that whenever the number of errors in the channel is bounded, and the graph expands sufficiently, a zero-valued dual solution to the decoding LP can be constructed. This dual solution implies that the transmitted codeword is optimal for the primal LP, and so LP decoding succeeds.

1.1 Comparison with Expander Codes

The original algorithm for expander codes, by Sipser and Spielman [23], has a theoretical performance guarantee similar to Theorem 1. In fact, when the expansion parameter δ equals $3/4$, the error-correction guarantee given in Theorem 1 for LP decoding matches the Sipser-Spielman bound exactly. The LP decoder — and other iterative algorithms such as sum-product and min-sum — have the advantage that they work in more general channels such as AWGN, and use "soft information" from the channel, at the expense of an increase in running time. However, our preliminary experiments indicate that even under the BSC, LP decoding (and these other iterative algorithms) perform significantly better than the bit-flipping algorithm in [23].

Remarkable progress toward the capacity of the BSC has been made by studying more general expander codes [23, 26, 2], where the "check nodes" are allowed to represent arbitrary linear sub-codes (as opposed to a parity code in the case of LDPC codes). The natural LP decoder for these codes is stronger than the one obtained by reducing the code to an LDPC code and applying the LP from [11]; therefore, the results in this paper should not be compared to these more powerful codes. However, many of the methods that we use in this paper still apply, and this topic merits further investigation. Other code constructions using expander graphs have been studied as well (e.g., [15]).

1.2 Outline

In Section 2, we provide background on LDPC codes, and the associated LP decoder from [11]. In Section 3, we show how to prove an error bound using the dual LP; it is worth noting that this method applies to any LP decoder, not just the one for LDPC codes. Section 4 is devoted to the proof of our main result using the expansion of the Tanner graph. In Section 5, we show that graphs with sufficient expansion exist and can be constructed efficiently.

2 An LP Decoder for Low-Density Parity-Check Codes

We begin by reviewing low-density parity-check (LDPC) codes and the LP decoder from [11]. Let $V = \{1, \dots, n\}$ and $C = \{1, \dots, m\}$ be indices for the columns (respectively rows) of the $m \times n$ parity check matrix H of a binary linear code $\mathcal{C}$ with rate at least $1 - m/n$. The *Tanner* or *factor* graph representation of the code $\mathcal{C}$ is a bipartite graph $\mathcal{G}$ with node sets V and C, and edges (i, j) for all i, j where $H_{j,i} = 1$. If the parity check matrix has a bounded number (independent of n) of non-zero entries in each column, we say that it has *low-density*; this condition translates to each node in V having bounded degree. In this paper, we do not require that the check nodes have bounded degree.

The code can be visualized directly from the graph $\mathcal{G}$. Imagine assigning to each variable node i a value in $\{0, 1\}$, representing the value of a particular code bit. A parity check node j is "satisfied" if the bits assigned to the variable nodes in its neighborhood have even parity (sum to zero mod 2). The n bits assigned to the variable nodes form a code word if and only if all check nodes are satisfied.

We assume that the graph $\mathcal{G}$ is left-regular; i.e., the degree of each variable node $i \in V$ is exactly some constant c. Let $N(S)$ denote the neighbors of a node set S. For a single node i, we let $N(i) := N(\{i\})$. For each check $j \in C$, let $\mathcal{E}_j := \{S \subseteq N(j) : |S| \text{ even}\}$. Each $S \in \mathcal{E}_j$ represents a "local codeword;" in other words, if we set each bit in S to 1, and other all bits in $N(j)$ to 0, then we satisfy check j.

Let γ_i be the "cost" of node i, where γ_i is the log-likelihood ratio for the i^{th} code bit. In the binary symmetric channel (BSC), we may assume that $\gamma_i = +1$ if a 0 is received from the channel for bit i, and $\gamma_i = -1$ if a 1 is received. We assume that the codeword 0^n is sent over the channel; this assumption is valid since the polytope for LDPC codes [11] is "$\mathcal{C}$-symmetric" (see [9]). Therefore, we have $\gamma_i = -1$ with probability p, and $\gamma_i = +1$ otherwise. For a particular setting of the cost vector γ, let $U = \{i \in V : \gamma_i = -1\}$ be the set of negative-cost variable nodes. The relevant setting of γ will be always clear from context.

2.1 The LP Decoder for LDPC Codes

The LP from [11] has an LP variable f_i for each node $i \in V$, indicating the value of the i^{th} code bit. In addition, for each parity check $j \in C$ and each set $S \in \mathcal{E}_j$ there is an LP variable $w_{j,S}$, which serves as an indicator for using the local codeword S to satisfy j. Note that the variable $w_{j,\emptyset}$ is also present for each parity check, and represents setting all bits in $N(j)$ to zero. We now give the decoding LP along with its dual, which we use in the next section:

Decoding LP:
$$\text{minimize} \quad \sum_i \gamma_i f_i \quad \text{s.t.}$$
$$\forall j \in C, \quad \sum_{S \in \mathcal{E}_j} w_{j,S} = 1$$
$$\forall \text{ edges } (i,j), \quad f_i = \sum_{S \in \mathcal{E}_j, S \ni i} w_{j,S}$$
$$\forall j \in C, \forall S \in \mathcal{E}_j, \ w_{j,S} \geq 0; \quad \forall i \in V, \ f_i \geq 0$$

Dual:
$$\text{maximize} \quad \sum_j v_j \quad \text{s.t.}$$
$$\forall j \in C, S \in \mathcal{E}_j, \quad \sum_{i \in S} \tau_{ij} \geq v_j \tag{1}$$
$$\forall i \in V, \quad \sum_{j \in N(i)} \tau_{ij} \leq \gamma_i \tag{2}$$
$$\forall j \in C, \ v_j \text{ free}; \quad \forall \text{ edges } (i,j), \ \tau_{ij} \text{ free}$$

Note that the constraints $f_i \leq 1$ and $w_{j,S} \leq 1$ are implied by the other constraints. Let w^0 be the setting of the w variables appropriate for when $f = 0^n$; i.e., for all $j \in C$ and $S \in \mathcal{E}_j$, we have $w_{j,S} = 1$ if $S = \emptyset$, and $w_{j,S} = 0$ otherwise.

The decoding algorithm works as follows. First, we solve the decoding LP to obtain an optimal solution (f^*, w^*). If $f^* \in \{0,1\}^n$, then f^* must represent the ML codeword [9]. In this case, we output f^*; otherwise, if some f_i^* has a fractional value, we declare an error. Our LP decoder will succeed if $(0^n, w^0)$ is the unique optimum solution of the LP. An important fact is that the decoding LP is solvable in polynomial time even if some of the check nodes have large degree; we refer the reader to [11, 8] for details.

3 Proving Error Bounds Using a Dual Feasible Point

In order to prove that LP decoding succeeds, we must show that $(0^n, w^0)$ is the unique optimum of the LP. To be conservative, we assume failure in the event that the LP has multiple optima, so that the LP decoder succeeds if and only if $(0^n, w^0)$ is the unique optimum solution. Consider the dual (given above) of the decoding LP. If there is a feasible point of the dual LP that has the same cost (i.e., zero) as the point $(0^n, w^0)$ has in the decoding LP, then $(0^n, w^0)$ is also an optimal point of the decoding LP. Therefore, to prove that the LP decoder succeeds, we could simply find a zero-cost point in the dual. Actually, since the existence of the zero-cost dual point only proves that $(0^n, w^0)$ is one of possibly many primal optima, we need to be a bit more careful; in particular, we give a dual feasible point that is strictly bounded away from its cost constraints, which implies using complementary slackness [22] that $(0^n, w^0)$ is the unique optimal solution to the LP. We make this argument precise in the upcoming proof.

We refer to the values τ_{ij} as "edge weights." The following definition is a useful characterization of zero-cost dual solutions:

Definition 1 *A setting of edge weights* $\{\tau_{ij}\}$ *is* feasible *if* **(i)** *for all checks* $j \in C$ *and distinct* $i, i' \in N(j)$, *we have* $\tau_{ij} + \tau_{i'j} \geq 0$, *and* **(ii)** *for all nodes* $i \in V$, *we have* $\sum_{j \in N(i)} \tau_{ij} < \gamma_i$.

Proposition 2 *If there is a feasible setting of edge weights, then the point* $(0^n, w^0)$ *is the* ***unique*** *optimum of the decoding LP.*

Proof: Let $\{\tau_{ij}\}$ be a feasible setting of edge weights. Taking $v_j = 0$ for all j gives a zero-cost dual solution; it is easily verified that this solution satisfies the dual constraints (1) and (2) by applying, respectively, conditions (i) and (ii) from Definition 1. (For (1), note that when $v_j = 0$, the constraint described by equation (1) is redundant for all S where $|S| \neq 2$.) It follows from the discussion above that $(0^n, w^0)$ is optimal for the cost function γ in the decoding LP.

We now show that $(0^n, w^0)$ is the unique optimum. The strict inequality in part (ii) of Definition 1 implies that $\sum_{j \in N(i)} \tau_{ij} \leq \gamma_i - \iota$ for some positive number ι, from which it follows that $(0^n, w^0)$ is an optimal point of the decoding LP under the cost function γ' where $\gamma'_i = \gamma_i - \iota$ for all i.

Now suppose $(0^n, w^0)$ is not the unique LP optimum under the original cost function γ. Since w^0 is the only feasible setting of the w variables when $f = 0^n$, there must be some other feasible point (f', w') where $f' \neq 0^n$ and $\sum_i \gamma_i f'_i = 0$. But since $f' \neq 0^n$, we have $\sum_i \gamma'_i f'_i < 0$, which contradicts the fact that $(0^n, w^0)$ is optimal under γ'. ■

4 Using Expansion to Give a Dual Feasible Point

In this section, we show how to assign weights τ_{ij} to each edge in the graph so we may apply Proposition 2. Below we define a special subset of edges called a δ-*matching*; this set is defined relative to the error pattern received from the channel. Our first step is to show that if a δ-matching exists, then we can find a feasible assignment of edge weights. We then prove that a δ-matching does indeed exist as long as the number of bits flipped by the channel is at most a constant fraction of n, where the constant depends on the expansion properties of the graph.

4.1 Definitions and notation

For the remainder of this section let $\mathcal{G}$ be a Tanner graph with n variable nodes each of degree c, and moreover let $\mathcal{G}$ be an $(\alpha n, \delta c)$-expander, where $\delta > 2/3 + 1/(3c)$ and δc is an integer. We also fix the following parameters, which are implicit functions of δ and/or the cost vector γ. Let $\lambda = 2(1 - \delta) + 1/c$. Note that $0 < \lambda < \delta$, and that λc is an integer. Recall that $U = \{i \in V : \gamma_i = -1\}$. Let $\dot{U}$ be the set of positive-cost variable nodes that have more than

$(1-\lambda)c$ neighbors in $N(U)$; i.e., $\dot{U} = \{i \in V : i \notin U,\ |N(i) \cap N(U)| \geq (1-\lambda)c + 1\}$. Let $U' = U \cup \dot{U}$.

Definition 2 *A δ-*matching *of U is a subset M of the edges incident to U' such that* **(i)** *every check in $N(U')$ is incident to at most one edge of M,* **(ii)** *every node in U is incident to at least δc edges of M, and* **(iii)** *every node in $\dot{U}$ is incident to at least λc edges of M.*

4.2 Assigning weights using a δ-matching

We give our weight assignment scheme in the following theorem. The existence of such an assignment implies decoding success, by Proposition 2.

Proposition 3 *If there is a δ-matching of U, then there is a feasible edge weight assignment.*

Proof: Call a check node j in C *activated* if j is incident to an edge (i, j) of M, and $i \in U$. Note that an activated check is incident to exactly one edge of M, by the definition of M. We assign edge weights as follows, using a positive constant x that we define later:

- For all activated checks j, we have $(i, j) \in M$ for some $i \in U$, and $(i', j) \notin M$ for all other $i' \in N(j)$. Set $\tau_{ij} = -x$, and set $\tau_{i'j} = +x$ for all other $i' \in N(j)$.

- For all other checks, set all incident edge weights to zero.

This weighting clearly satisfies condition (i) of a feasible weight assignment. For condition (ii), we distinguish three cases. For the following argument, note that all edges in M incident to nodes in U receive weight $-x$, all other edges in M receive weight 0, and all edges not in M receive weight either $+x$ or 0.

1. For a variable node $i \in U$, we have $\gamma_i = -1$. Also, at least δc of the edges incident to i are in M (and each has weight $-x$). All other incident edges have weight either $+x$ or 0. In either case, each has weight at most $+x$, and so the total weight of incident edges is at most $\delta c(-x) + (1-\delta)cx = (1-2\delta)cx$. This is less than -1 as long as $x > \frac{1}{(2\delta-1)c}$.

2. If $i \in \dot{U}$, then $\gamma_i = +1$. At least λc of i's incident edges are in M, but not incident to U; these edges have weight 0. All other incident edges have weight either $+x$ or 0. In either case, they each have weight at most $+x$, and so the total weight of incident edges is at most $(1-\lambda)cx$, which is less than $+1$ as long as $x < \frac{1}{(1-\lambda)c}$.

3. The remaining case is when $i \notin U'$, and in this case $\gamma_i = +1$. The definition of $\dot{U}$ implies that i has at least λc neighbors not in $N(U)$, and so at most $(1-\lambda)c$ edges incident to i have non-zero weight. We are therefore in the same situation as in the previous case: all non-zero weights are at most $+x$, and so the total weight of incident edges is at most $(1-\lambda)cx$, which is less than $+1$ as long as $x < \frac{1}{(1-\lambda)c}$.

Summarizing our conditions on x, we have $\frac{1}{(2\delta-1)c} < x < \frac{1}{(1-\lambda)c}$. There is a feasible x satisfying these conditions as long as $(1-\lambda) < (2\delta-1)$, which is true by the definition of λ. ■

4.3 Expansion implies a δ-matching

To construct our feasible weight assignment, it remains to show that we can construct a δ-matching. To do so, we use the expansion of the graph.

Proposition 4 *If $\mathcal{G}$ is an $(\alpha n, \delta c)$-expander with $\delta > 2/3 + 1/(3c)$, and $|U'| \leq \alpha n$, then U has a δ-matching.*

Proof: We construct the δ-matching M by setting up a max-flow instance (see [5], [1]). We will construct this flow instance using the variable nodes U', the check nodes $N(U')$, and directed versions of the edges incident to U'. We will also introduce two new nodes (a source and a sink), as well as edges incident to those nodes.

We construct the flow instance as follows, with all integer capacities: For every edge (i, j) in $\mathcal{G}$ where $i \in U'$ and $j \in N(U')$, make a directed edge $i \rightarrow j$ with capacity 1. Create a source s, and make a new edge with capacity δc from s to every variable node. Create a sink t, and make a new edge with capacity 1 from each check node to t.

We claim that if there exists a flow of value $\delta c|U'|$ in this instance, then there is a δ-matching M. Let f be a flow of value $\delta c|U'|$; without loss of generality we may assume f is integral [1]. We set M to be the set of original edges (from U' to $N(U')$) with unit flow in f. Since f has value $\delta c|U'|$, every edge out of the source s to the nodes of U' must be saturated. It follows that exactly δc edges out of each $i \in U'$ have a unit of flow in f. Thus M satisfies condition (ii) of a δ-matching, and since $\lambda < \delta$, the set M is more than sufficient to satisfy condition (iii) as well. The edges from each check $j \in N(U')$ to the sink have capacity 1, and so at most one incoming edge to each check is carrying flow in f. It follows that at most one edge of M is incident to each check in $N(U')$, and thus M satisfies condition (i) of a δ-matching.

So it remains to show that there exists a flow of value $\delta c|U'|$, or equivalently [1] that the minimum s-t cut is at least $\delta c|U'|$. Let $(\mathrm{V_s}, \mathrm{C_s}, \mathrm{V_t}, \mathrm{C_t})$ describe the min-cut as follows: $\mathrm{V_s}$ and $\mathrm{C_s}$ are the variable and check nodes, respectively, on the same side of the cut as the source s. Similarly, $\mathrm{V_t}$ and $\mathrm{C_t}$ are the variable and check nodes, respectively, on the the same side of the cut as the sink t.

Lemma 5 *Without loss of generality, there are no edges in the min-cut from* $\mathrm{V_s}$ *to* $\mathrm{C_t}$*.*

Proof: Consider an edge (i, j), where $i \in \mathrm{V_s}$ and $j \in \mathrm{C_t}$. If we move j to the source side of the cut, then we add at most 1 to the cut value, since the only edge leaving j is the one to the sink t. However, we also subtract at least 1 from the cut value, because the edge (i, j) is no longer in the cut. □

For node sets A and B, let $[A, B]$ denote the total capacity of edges going from A to B. The value of the min-cut is exactly $[\{s\}, \mathrm{V_t}] + [\mathrm{C_s}, \{t\}] + [\mathrm{V_s}, \mathrm{C_t}]$. Note that $[\{s\}, \mathrm{V_t}] = \delta c|\mathrm{V_t}|$, and $[\mathrm{C_s}, \{t\}] = |\mathrm{C_s}|$. Furthermore, Lemma 5 implies $[\mathrm{V_s}, \mathrm{C_t}] = 0$ and $\mathrm{C_s} \supseteq N(\mathrm{V_s})$. So, we have that the min-cut has value

$$\begin{aligned} \delta c|\mathrm{V_t}| + |\mathrm{C_s}| &\geq \delta c|\mathrm{V_t}| + |N(\mathrm{V_s})| && \text{(3a)} \\ &\geq \delta c|\mathrm{V_t}| + \delta c|\mathrm{V_s}| && \text{(3b)} \\ &= \delta c|U'|, \end{aligned}$$

where (3a) follows from $\mathrm{C_s} \supseteq N(\mathrm{V_s})$, and (3b) follows from the expansion of G. ■

4.4 Proof of our main theorem

Before proceeding to the proof of Theorem 1, we require the following:

Lemma 6 *Suppose $|U| \leq \frac{\alpha n - 1}{1+\beta}$, where $\beta = \frac{1-\delta}{3\delta - 2}$. Then, we have $|\dot{U}| \leq \beta|U|$.*

Proof: Suppose not. Then there is some subset $\ddot{U} \subseteq \dot{U}$ where $|\ddot{U}| = \lfloor \beta|U| \rfloor + 1$. Consider the set $U \cup \ddot{U}$. Since $|U + \ddot{U}| = |U| + \lfloor \beta|U| \rfloor + 1 \leq |U|(1+\beta) + 1$, we have $|U + \ddot{U}| \leq \alpha n$ by our assumption on $|U|$. Therefore this set expands, and we have **(i):** $|N(U \cup \ddot{U})| \geq c\delta(|U| + |\ddot{U}|)$.

Furthermore, we have $|N(U \cup \ddot{U})| = |N(U)| + |N(\ddot{U}) \setminus N(U)| \leq c|U| + |N(\ddot{U}) \setminus N(U)|$. Consider the set $N(\ddot{U}) \setminus N(U)$. These are the edges from $\ddot{U}$ that are not incident to $N(U)$. Each node in $\ddot{U}$ has at most $\lambda c - 1$ such edges, by the definition of $\dot{U}$. Therefore, $|N(\ddot{U}) \setminus N(U)| \leq (\lambda c - 1)|\ddot{U}|$, and we have **(ii):** $|N(U \cup \ddot{U})| \leq c|U| + (\lambda c - 1)|\ddot{U}|$.

Combining the inequalities **(i)** and **(ii)** we get $|\ddot{U}| \leq \frac{(1-\delta)c}{(\delta - \lambda)c + 1}|U| = \beta|U|$, a contradiction. ■

We are now ready to prove our main theorem:

Theorem 1 *Let $\mathcal{C}$ be a low-density parity-check code with length n and rate at least $1 - m/n$ described by a Tanner graph $\mathcal{G}$ with n variable nodes, m check nodes, and regular left degree c. Suppose $\mathcal{G}$ is an $(\alpha n, \delta c)$-expander, where $\delta > 2/3 + 1/(3c)$ and δc is an integer. Then the LP decoder succeeds, as long at most $\frac{3\delta-2}{2\delta-1}(\alpha n - 1)$ bits are flipped by the channel.*

Proof: By assumption, $|U| \leq \frac{3\delta-2}{2\delta-1}(\alpha n - 1) = \frac{\alpha n - 1}{1+\beta}$, and so by Lemma 6 we have $|\dot{U}| \leq \frac{1-\delta}{3\delta-2}|U|$. This implies $|U'| = |U| + |\dot{U}| \leq \alpha n$. Therefore, by Proposition 4, there exists a δ-matching of U, and so by Proposition 3 there exists a feasible weight assignment. Using Proposition 2, we conclude that $(0^n, w^0)$ is the unique optimum of the LP, and so the decoder succeeds. ■

For any constant rate between 0 and 1, a random graph will meet the conditions of the above theorem for some δ as required and some constant $\alpha > 0$; also explicit families of such graphs can be constructed efficiently (we discuss this more in the next section). As an example of Theorem 1, let us set $\delta = 3/4$. Using an $(\alpha n, 3c/4)$-expander, Theorem 1 asserts that the LP decoder will succeed if fewer than $\alpha n/2$ bits are flipped by the channel. Interestingly, this result matches the parameters of the statement given by Sipser and Spielman [23] in the original paper on expander codes (decoding success if fewer than $\alpha n/2$ errors using a $(\alpha n, 3c/4)$-expander).

5 Existence, Construction of Expanders

5.1 Expansion from Random Graphs

Using the probabilistic method one can show the following:

Proposition 7 *Let $0 < r < 1$ and $0 < \delta < 1$ be any fixed constants, and let c be such that $(1-\delta)c$ is an integer which is at least 2. Then for any n, m such that $r = 1 - \frac{m}{n}$ there is a Tanner graph with n variable nodes, m check nodes, and regular left degree c which is an $(\alpha n, \delta c)$-expander, where*

$$\alpha = \left(2e^{\delta c+1}\left(\delta c/(1-r)\right)^{(1-\delta)c}\right)^{-\frac{1}{(1-\delta)c-1}}. \tag{4}$$

Proof: We consider random (n, m)-bipartite graphs which are formed as follows:

- For $i = 1, \ldots, n$ the i-th variable node uniformly picks a c-element subset of $[m]$ and forms edges to these check nodes.

Any graph formed this way is c-regular on the left. We let d denote cn/m, the average degree of the check nodes.

We note first that each set consisting of a single variable node clearly expands by a factor of exactly c.

Now fix a value $s \geq 2$, a set S of left-vertices where $|S| = s$, and a set T of right-vertices of size $t := \delta cs$ (note that t is an integer). For each individual vertex in S, the probability that all c of its neighbors lie in T is

$$\frac{t}{m} \cdot \frac{t-1}{m-1} \cdots \frac{t-c+1}{m-c+1} < \left(\frac{t}{m}\right)^c.$$

Since each left-vertex chooses its neighbors independently of the other left-vertices, the probability that $N(S) \subseteq T$ is at most $\left(\frac{t}{m}\right)^{cs}$. Since there are $\binom{n}{s}$ sets S of s left-vertices and $\binom{m}{t}$ sets T of $t = \delta cs$ right-vertices, the probability that *any* set of s left-vertices has its neighborhood of size at most t is at most

$$\begin{aligned}
\binom{n}{s}\binom{m}{t}\left(\frac{t}{m}\right)^{cs} &\leq \left(\frac{en}{s}\right)^s \left(\frac{em}{t}\right)^t \left(\frac{t}{m}\right)^{cs} \\
&= \left(\frac{en}{s}\right)^s \left(\frac{em}{\delta cs}\right)^{\delta cs} \left(\frac{\delta cs}{m}\right)^{cs} \\
&= \left(\frac{en}{s}\right)^s \left(\frac{en}{\delta ds}\right)^{\delta cs} \left(\frac{\delta ds}{n}\right)^{cs} \\
&= \left[(\delta d)^{(1-\delta)c}\, e^{\delta c+1} \left(\frac{s}{n}\right)^{(1-\delta)c-1}\right]^s. \qquad (5)
\end{aligned}$$

Let $K = (\delta d)^{(1-\delta)c} e^{\delta c+1}$ and $a = (1-\delta)c - 1$, so (5) equals $\left[K\left(\frac{s}{n}\right)^a\right]^s$. It is easily checked that for $s \leq n/(2K)^{1/a}$, the quantity $K\left(\frac{s}{n}\right)^a$ is at most $1/2$, and thus we have

$$\sum_{s=2}^{n/(2K)^{1/a}} (5) \leq \sum_{s=2}^{n/(2K)^{1/a}} 1/2^s < \sum_{s=2}^{\infty} 1/2^s = 1/2.$$

Thus with probability at least $1/2$, we have that a random graph G formed as described above is an $(\alpha n, \delta c)$-expander for $\alpha = 1/(2K)^{1/a}$. Plugging in for K and a and recalling that $d = c/(1-r)$ the proposition is proved. ∎

Together with Theorem 1, Proposition 7 implies that there are LDPC codes of any constant rate for which LP decoding corrects a constant fraction of error. As a concrete example, if we take $r = 1/2$, $\delta = \frac{3}{4}$, and $c = 36$, we have that there is a family of LDPC codes of rate $1/2$ for which LP decoding can correct .000155 fraction of errors.

We note that a more careful analysis of the random bipartite graphs used to prove Proposition 1 gives a stronger bound on α, but this bound does not have a convenient closed form. Using this stronger bound it can be shown that for the specific family of LDPC codes described above (with $r = 1/2$, $\delta = 3/4$ and $c = 36$) LP decoding can correct .000175 fraction of errors.

To see this, note that the proof of Proposition 1 implies that the probability (over our choice of a random graph) that any set of size up to αn fails to expand is at most $1/2$, where α is defined in equation (4). Using a different bound on binomial coefficients we can show that for some $\alpha' > \alpha$ to be described below, all sets of of size $s = \hat{\alpha} n$ fail to expand with exponentially low probability, where $\hat{\alpha}$ is any fixed constant value in the open interval $(0, \alpha')$. Combining these facts, we have that a random G is a $(\alpha' n, \delta c)$-expander with probability at least $1/2 - o(1)$.

We now simply use the "entropy bound"

$$\binom{n}{\hat{\alpha} n} \leq 2^{(H(\hat{\alpha}) + o(1))n}$$

for binomial coefficients in our bound instead of the $\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b$ bound. Using $s = \hat{\alpha} n$, $m = nc/d$ and $t = \delta cs$ this gives

$$\begin{aligned} \binom{n}{s}\binom{m}{t}\left(\frac{t}{m}\right)^{cs} &\leq 2^{[H(\hat{\alpha})+o(1)]n} \cdot 2^{\frac{c}{d}[H(\delta\hat{\alpha}d)+o(1)]n} \cdot (\delta\hat{\alpha}d)^{c\hat{\alpha}n} \\ &= 2^{[H(\hat{\alpha})+\frac{c}{d}H(\delta\hat{\alpha}d)+c\hat{\alpha}\log_2(\delta\hat{\alpha}d)+o(1)]n}. \end{aligned}$$

Thus if $\hat{\alpha}$ is any constant value such that

$$H(\hat{\alpha}) + \frac{c}{d} H(\delta\hat{\alpha}d) + c\hat{\alpha}\log_2(\delta\hat{\alpha}d) < 0 \tag{6}$$

we then have that with probability $2^{-\Theta(n)}$ all sets of size $s = \hat{\alpha} n$ satisfy the required expansion. Inequality (6) does not seem to yield a nice closed form expression for $\hat{\alpha}$. However, one can verify that e.g. for $c = 36$, $d = 72$, and $\delta = \frac{3}{4}$, any value $0 < \hat{\alpha} \leq .00035$ causes (6) to be negative. This gives the stronger LP decoding performance bound claimed earlier.

5.2 Explicit Constructions of Expanders

Recently, Capalbo *et al.* [4] gave the first explicit construction of *lossless* expanders (namely with δ arbitrarily close to 1), using the zig-zag graph product [19] through the framework of randomness conductors. Their work implies the following.

Proposition 8 *Let $0 < r < 1$ and $0 < \delta < 1$ be any fixed constants. Then for any n, m such that $r = 1 - \frac{m}{n}$ there is an efficiently constructible Tanner graph with n variable nodes, m check nodes, and regular left degree c which is an $(\alpha n, \delta c)$-expander, where $c = poly(\log(1-r), 1/(1-\delta))$, and $\alpha = \Omega((1-\delta)(1-r)/c)$.*

Thus, there are efficiently constructible LDPC codes of any constant rate for which LP decoding corrects a constant fraction of errors. Note that while the above proposition does not directly guarantee δc to be an integer, this is not a problem since given any value for δ there is some δ' such that $\delta - 1/c \leq \delta' \leq \delta$ and $\delta' c$ is an integer (note that any $(\alpha n, \delta c)$-expander is clearly also an $(\alpha n, \delta' c)$-expander for any $\delta' \leq \delta$). Thus, in order to apply Theorem 1, it is sufficient to choose some $\delta > 2/3 + 1/(3c) + 1/c$ for the Capalbo *et al.* construction.

6 Conclusions

We have given the first strong word error rate bound for LP decoding; furthermore this bound is better than any finite-length bound known for the conventional message-passing decoders.

The next logical step is to adapt our techniques to different codes and channels. The idea of constructing a dual solution with value zero to prove decoding success applies to any "$\mathcal{C}$-symmetric" [9] LP decoder and memoryless symmetric channel (such as the AWGN channel). In fact for LDPC codes, the definition of a feasible assignment of edge weights (along with Proposition 2) can be used verbatim for any memoryless symmetric channel.

Acknowledgments

We would like to thank G. David Forney Jr., David Karger, Ralf Koetter and Pascal Vontobel for helpful discussions.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice-Hall, 1993.

[2] A. Barg and G. Zémor. Error exponents of expander codes. *IEEE Transactions on Information Theory*, 48(6):1725–1729, 2002.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: turbo-codes. *Proc. IEEE International Conference on Communication (ICC), Geneva, Switzerland*, pages 1064–1070, May 1993.

[4] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree/2 barrier. In *Proc. 34th ACM Symposium on the Theory of Computing*, pages 659–668, 2002.

[5] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. M.I.T. Press, Cambridge, Massachusetts, U.S.A., 2001.

[6] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke. Finite length analysis of low-density parity check codes. *IEEE Transactions on Information Theory*, 48(6), 2002.

[7] G. Even and N. Halabi. Improved bounds on the word error probability of RA(2) codes with linear programming based decoding. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[8] J. Feldman. *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Technology, 2003.

[9] J. Feldman, D. R. Karger, and M. J. Wainwright. LP decoding. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[10] J. Feldman and David R. Karger. Decoding turbo-like codes via linear programming. *Proc. 43rd annual IEEE Symposium on Foundations of Computer Science (FOCS)*, November 2002. To appear in *Journal of Computer and System Sciences*.

[11] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode linear codes. *37th annual Conference on Information Sciences and Systems (CISS '03)*, March 2003. Submitted to *IEEE Transactions on Information Theory*, May, 2003.

[12] G. D. Forney, R. Koetter, F. R. Kschischang, and A. Reznik. On the effective weights of pseudocodewords for codes defined on graphs with cycles. In *Codes, systems and graphical models*, pages 101–112. Springer, 2001.

[13] B. Frey, R. Koetter, and A. Vardy. Signal-space characterization of iterative decoding. *IEEE Transactions on Information Theory*, 47(2):766–781, 2001.

[14] R. Gallager. Low-density parity-check codes. *IRE Trans. Inform. Theory*, IT-8:21–28, Jan. 1962.

[15] V. Guruswami and P. Indyk. Expander-based constructions of efficiently decodable codes. *42nd Symposium on Foundations of Computer Science (FOCS)*, 2001.

[16] R. Koetter. Personal communication, 2002.

[17] R. Koetter and P. O. Vontobel. Graph-covers and iterative decoding of finite length codes. In *Proc. 3rd International Symposium on Turbo Codes*, September 2003.

[18] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Improved low-density parity-check codes using irregular graphs and belief propagation. *Proc. 1998 IEEE International Symposium on Information Theory*, page 117, 1998.

[19] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proc. 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 3–13, 2000.

[20] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity approaching irregular low-density parity check codes. *IEEE Transactions on Information Theory*, 47:619–637, Feb. 2001.

[21] T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2), February 2001.

[22] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley, 1987.

[23] M. Sipser and D. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[24] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[25] N. Wiberg. *Codes and Decoding on General Graphs.* PhD thesis, Linkoping University, Sweden, 1996.

[26] G. Zémor. On expander codes. *IEEE Transaction of Information Theory*, 47(2):835–837, 2001.