LP Decoding Corrects a Constant Fraction of Errors

Jon Feldman, Tal Malkin, Rocco A. Servedio, Cliff Stein, and Martin J. Wainwright, *Member, IEEE*

***Abstract*—We show that for low-density parity-check (LDPC) codes whose Tanner graphs have sufficient expansion, the linear programming (LP) decoder of Feldman, Karger, and Wainwright can correct a constant fraction of errors. A random graph will have sufficient expansion with high probability, and recent work shows that such graphs can be constructed efficiently. A key element of our method is the use of a *dual witness*: a zero-valued dual solution to the decoding linear program whose existence proves decoding success. We show that as long as no more than a certain constant fraction of the bits are flipped by the channel, we can find a dual witness. This new method can be used for proving bounds on the performance of any LP decoder, even in a probabilistic setting. 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 error bound for LDPC codes using an analysis based on "pseudocodewords." Recent work by Koetter and Vontobel 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 that there exist families of LDPC codes where the minimum BSC pseudoweight grows linearly in the block length.**

***Index Terms*—Channel coding, factor graphs, iterative decoding, linear programming, low-density parity-check (LDPC) codes, message passing, Tanner graphs.**

I. Introduction

Turbo codes [7] and low-density parity-check (LDPC) codes [8] 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* [9]–[11]. 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 [12].

Successful finite-length analysis of a suboptimal 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.* [13] identified the message-passing pseudocodewords as "stopping sets," and used this characterization to give decoding performance results. For more general channels, Wiberg [14] (see also [15], [16]) used the computation tree associated with 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 [3], [1], [2] provides an alternative to message-passing decoding that is more amenable to finite-length analysis. Specific LP decoders have been defined for turbo codes [17] and LDPC codes [1], [2], [18]. The pseudocodewords for an LP decoder are the vertices of a 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 [3]). For the case of LDPC codes, Koetter and Vontobel [5], [6] described these LP pseudocodewords as codewords in "graph covers," and established a connection to the pseudocodewords of the message-passing "min-sum" algorithm. For general factor graphs, Wainwright *et al.* [19], [20] connected tree-based LP relaxations (such as the LP decoder [2]) with the broader class of reweighted min-sum algorithms.

Until now, the only known performance results for the LP decoder were based on graphs with high girth; in particular, exploiting this high girth yielded an inverse-polynomial word-error rate (WER) bound for rate-$1/2$ repeat–accumulate codes [17], [21], [22], 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 errors in bit-flipping channels. Using a simple Chernoff bound, this fact implies WER $\leq e^{-\Omega(n)}$ under the binary-symmetric channel (BSC). This result constitutes the first proof that LP decoding has an inverse-exponential WER on a constant-rate code. Furthermore, no such WER 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. More specifically, a Tanner graph $\mathcal{G}$ is a

Manuscript received February 22, 2005; revised April 15, 2006. The work of J. Feldman is supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship DMS-0303407. The work of T. Malkin is supported in part by the National Science Foundation CAREER award CCF-0347839. The work of R. A. Servedio is supported in part by the National Science Foundation CAREER award CCF-0347282. The work of C. Stein is supported in part by the National Science Foundation under Grant DMS-9970063 and an Alfred P. Sloan Foundation Fellowship. The work of M. J. Wainwright is supported in part by the National Science Foundation under Grant DMS-0528488 and an Alfred P. Sloan Foundation Fellowship. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, Jun./Jul. 20024.

J. Feldman was with the Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027 USA. He is now with Google, Inc., New York, NY 10016 USA. (e-mail: jonfeld@ieor.columbia.edu).

T. Malkin and R. A. Servedio are with the Department of Computer Science, Columbia University, New York, NY 10027-7003 USA (e-mail: tal@cs.columbia.edu;rocco@cs.columbia.edu).

C. Stein is with the Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027 USA (e-mail: cliff@ieor.columbia.edu).

M. J. Wainwright is with the Department of Electrical Engineering and Computer Science, and the Department of Statistics, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: wainwrig@eecs.berkeley.edu).

Communicated by A Ashikhmin, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.887523

(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 an LDPC 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 a $(\alpha n, \delta c)$-expander, where $\delta > 2/3 + 1/(3c)$ and δc is an integer. 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.

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, we can find a *dual witness*: a zero-valued dual solution to the decoding LP. This dual solution implies that the transmitted codeword is optimal for the primal LP, and so LP decoding succeeds.

A preliminary version of this paper was presented at the ISIT 2004 symposium [23]. We also note that some techniques developed in this paper have since been used to obtain results for more general expander codes [24].

A. Related Work: LDPC Codes

The bit-flipping algorithm for expander codes developed by Sipser and Spielman [25] 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. Burshtein and Miller [26] also use graph expansion to analyze the performance of various related iterative algorithms. With respect to the fraction of errors corrected on regular graphs, our results are roughly the same as theirs; specifically, for δ close to 1, both results show the ability to correct (close to) αn errors, where α is the expansion parameter. Lentmaier *et al.* [27] analyze iterative coding for LDPC codes (as well as other ensembles), and establish that under suitable technical conditions, the error probability decays as $\exp(-an^{\gamma})$ for some constant $\gamma < 1$.

The LP decoder—as well as iterative algorithms such as sum-product and min-sum—have the advantage that they apply in more general settings such as the additive white Gaussian noise (AWGN) channel, and exploit "soft information" from the channel. Although this entails an increase in running time, our preliminary experiments indicate that even for the binary-symmetric case, LP decoding (and other algorithms that exploit soft information) perform significantly better than the bit-flipping algorithm of Sipser and Spielman [25].

B. Related Work: Other Codes Built From Expanders

Expanders also play a role in generalized LDPC constructions. Zémor [28] and Barg and Zémor [29]–[31] have a series of papers analyzing expander codes where the "check nodes" are allowed to represent arbitrary linear subcodes (as opposed to single parity checks in the case of a standard LDPC code). They show that such codes, together with efficient message-passing algorithms, can achieve the capacity of the BSC, as well as correct adversarial errors up to (and beyond) the Zyablov bound. Guruswami and Indyk [32] (as well as Roth and Skatchek [33]) construct "near-MDS" codes using expanders. In later work [34], they achieve the Gilbert–Varsharmov (GV) bound for low rates.

The natural LP decoder for these codes is stronger than the one obtained by reducing the code to an LDPC code and applying the tree-based LP [1], [2] to the associated factor graph; therefore, the results in this paper should not be compared to these more powerful codes. Furthermore, since the preliminary version of this work [23], we have shown that LP decoding can achieve channel capacity [24], using a family of expander codes along the lines of [29]. While these more sophisticated expander-based constructions yield stronger theoretical bounds on error correction than those known for LDPC codes, the codes themselves are mostly impractical for use in communication systems, due to their dependence on large subcodes. (The size of these subcodes is often exponentially large in $1/\epsilon$, where ϵ is the gap between the code rate and the desired bound.) Therefore, the study of LDPC codes is of independent interest.

C. Outline

The remainder of this paper is organized as follows. In Section II, we provide background on LDPC codes, and the associated LP decoder from [2]. In Section III, we show how to prove an error bound using a dual witness; it is worth noting that this method applies to any LP decoder, not just the one for LDPC codes. Section IV is devoted to the proof of our main result using the expansion of the Tanner graph. We conclude with some final remarks and open questions in Section V. In the Appendix , we show that graphs with sufficient expansion exist and can be constructed efficiently.

II. Background

We begin by providing background on LDPC codes, as well as LP decoding applied to them [2].

A. LDPC Codes

Let $V = \{1, \ldots, n\}$ and $C = \{1, \ldots, m\}$ 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) between variable node i and check node j for all i, j where $H_{j,i} = 1$. If the parity-check matrix has a bounded number (independent of n) of nonzero 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 $\bmod 2$). The n bits assigned to the variable nodes form a codeword 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 all other 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 ith code bit. When transmitting over the BSC with crossover probability $p \in (0, \frac{1}{2})$, we may rescale the log-likelihood ratios such 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 [2] is "$\mathcal{C}$-symmetric" for any binary-input output-symmetric (BIOS) channel; see Feldman *et al.* [3] for further details. Therefore, for the BSC with crossover probability p, we have $\gamma_i = -1$ with probability p, and $\gamma_i = +1$ otherwise. For a particular setting of the cost vector γ, let $U = \{i \in V \mid \gamma_i = -1\}$ be the set of negative-cost variable nodes.

B. The LP Decoder for LDPC Codes

The first-order LP decoder for LDPC codes [2] has an LP variable f_i for each node $i \in V$, indicating the value of the ith 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 :

$$\text{minimize} \quad \sum_i \gamma_i f_i \quad \text{such that}$$

$$\forall j \in C, \quad \sum_{S \in \mathcal{E}_j} w_{j,S} = 1 \tag{1a}$$

$$\forall \text{ edges } (i,j), \quad f_i = \sum_{S \in \mathcal{E}_j, S \ni i} w_{j,S} \tag{1b}$$

$$\forall j \in C, \forall S \in \mathcal{E}_j,\ w_{j,S} \geq 0, \qquad \forall i \in V,\ f_i \geq 0. \tag{1c}$$

Dual LP :

$$\text{maximize} \quad \sum_j v_j \quad \text{such that}$$

$$\forall j \in C, S \in \mathcal{E}_j, \quad \sum_{i \in S} \tau_{ij} \geq v_j \tag{2a}$$

$$\forall i \in V, \quad \sum_{j \in N(i)} \tau_{ij} \leq \gamma_i \tag{2b}$$

$$\forall j \in C,\ v_j \quad \text{free} \qquad \forall \text{ edges } (i,j), \quad \tau_{ij} \text{ free.} \tag{2c}$$

Note that the constraints $f_i \leq 1$ and $w_{j,S} \leq 1$ are implied by the other constraints, assuming that every bit is connected to at least one check. 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 maximum-likelihood (ML) codeword [3]. 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. (We remind the reader of our previous assumption that we are sending the all-zeros codeword 0^n.) 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 the papers [2], [18] for details.

III. Proving Error Bounds Using a Dual Witness

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 of the decoding LP given above. 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, using standard results on LP duality [35], in order to prove that the LP decoder succeeds, it suffices to exhibit 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 (2b), which implies using complementary slackness [35] that $(0^n, w^0)$ is the unique optimal solution to the LP. We call such a dual point a *dual witness*. This argument is made precise in the upcoming proof.

We refer to the values τ_{ij} as *edge weights*. The following definition underlies a sufficient condition for a unique zero-cost dual solution:

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 (2a) and (2b) by applying, respectively, conditions i) and ii) from Definition 1. (For (2a), note that when $v_j = 0$, the constraint described by (2a) is redundant for all S where $|S| \neq 2$.) It follows from the preceding discussion 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'. □

The preceding result can easily be generalized to any LP decoder, where the dual witness takes on a different form depending on the structure of the code and the LP relaxation. In fact, a variant of this idea was explored in previous work [36] in the context of turbo codes, and has recently been explored for more general expander codes [24].

At one level, trying to find a dual witness is simply a reformulation of the problem of trying to prove that the transmitted codeword is the optimal primal LP solution. The value of looking at the dual lies in the analytical flexibility that it affords—in particular, an ability to trade off error bound quality for ease of analysis. Take, for example, the extreme case in which the channel is noiseless, so that $\gamma_i = +1$ for all i. In this case, finding a dual witness reduces to finding a feasible setting of edge weights, and is very easy: simply set all $\tau_{ij} = 0$. In general, as the noise increases (and hence more bits get flipped), it becomes increasingly difficult to find a dual witness.

IV. Using Expansion to Find a Dual Witness

This section is devoted to the proof of our main result, previously stated as Theorem 1. We begin by defining a procedure for assigning feasible edge weights τ_{ij}, which then allows us to apply Proposition 2. Our procedure uses a special subset of edges called a (δ, λ)-*matching*, defined in Section IV-A. The (δ, λ)-matching is a function of the error pattern received from the channel. In Section IV-B, we show that if a (δ, λ)-matching exists, then we can find a feasible assignment of edge weights. In Section IV-C, we prove that a (δ, λ)-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. We finish the proof of Theorem 1 in Section IV-D.

A. Definition 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 and sets, 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. Define $U := \{i \in V : \gamma_i = -1\}$, and let $\dot{U}$ be the set of positive-cost variable nodes outside U that have more than $(1-\lambda)c$ neighbors in $N(U)$ (i.e., $\dot{U} := \{i \in V : i \notin U,\ |N(i) \cap N(U)| \geq (1-\lambda)c + 1\}$). Finally, we define $U' := U \cup \dot{U}$.

Definition 2: A (δ, λ)-*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.

B. Assigning Weights Using a (δ, λ)-Matching

We give our weight assignment scheme in the following theorem (also in Fig. 1). The existence of such an assignment implies decoding success, by Proposition 2.

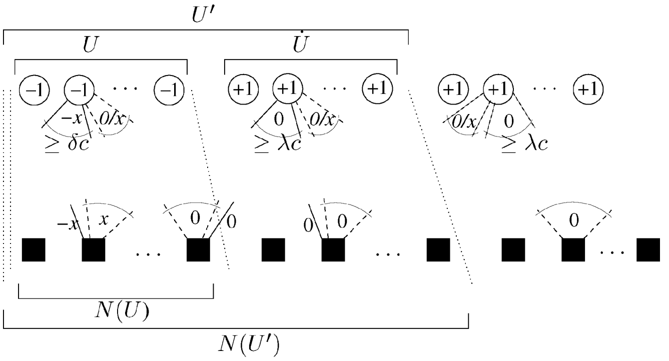

Fig. 1. A weighting scheme that satisfies the LP dual constraints. Given an error set $U = \{i \in V \mid \gamma_i = -1\}$, we let $\dot{U} \subseteq V$ be the nodes not in U with more than $(1-\lambda)c$ neighbors in $N(U)$, and let $U' = U \cup \dot{U}$. The matching M (solid edges) contains at most one edge incident to each check node, at least δc edges incident to each node in U, and at least λc edges incident to each node in $\dot{U}$. For all $(i, j) \in M$ such that $i \in U$, we set $\tau_{ij} = -x$, and $\tau_{i'j} = x$ for all $i' \neq i$. For all other j, we set all $\tau_{ij} = 0$.

Proposition 3: If there is a (δ, λ)-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 (see also Fig. 1), 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 (trivially) 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 nonzero weight. We are therefore in the same situation as in the previous case: all nonzero 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

$$\frac{1}{(2\delta-1)c} < x < \frac{1}{(1-\lambda)c}.$$

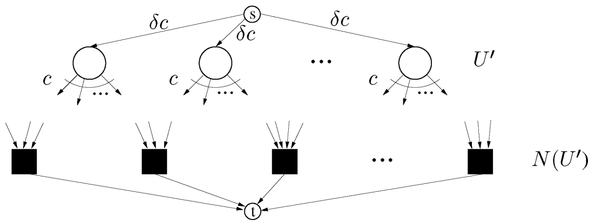

Fig. 2. An instance of max-flow used to show that graph expansion implies a (δ, λ)-matching (Proposition 4). All edges have unit capacity, except the edges leaving the source s, which have capacity δc.

There is a feasible x satisfying these conditions as long as $(1-\lambda) < (2\delta - 1)$, which is true by the definition of λ. □

C. Expansion Implies a (δ, λ)-Matching

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

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

Proof: We construct the (δ, λ)-matching M by setting up a max-flow instance (see the books [37], [38] for background on max-flow). 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 (see Fig. 2), 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 $j \in N(U')$ to t.

We claim that if there exists a flow of value $\delta c|U'|$ in this instance, then there is a (δ, λ)-matching M. Let f be a flow of value $\delta c|U'|$; without loss of generality, we may assume f is integral [38]. 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, λ)-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, λ)-matching.

So it remains to show that there exists a flow of value $\delta c|U'|$, or equivalently [38] 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 minimum s-t cut as follows: V_s and C_s are the variable and check nodes, respectively, on the same side of the cut as the source s. Similarly, V_t and C_t are the variable and check nodes, respectively, on the the same side of the cut as the sink t. This minimum s-t cut is depicted in Fig. 3.

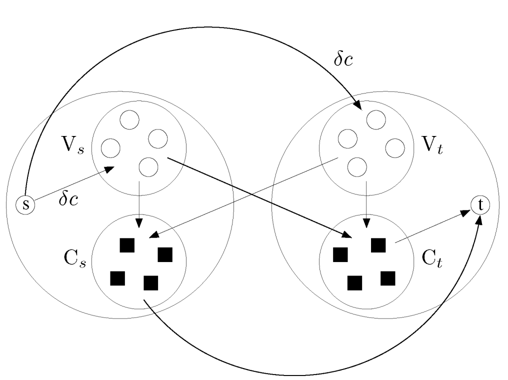

Fig. 3. A minimum s-t cut $(\mathrm{V}_s, \mathrm{C}_s, \mathrm{V}_t, \mathrm{C}_t)$ in the flow graph of Fig. 2. One of each edge type is shown, and the edges that contribute to the size of the cut are shown with solid lines. Every edge has unit capacity except those leaving the source s, which have capacity δc.

For node sets A and B, let $[A, B]$ denote the total capacity of edges going from A to B. The value of the minimum s-t 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 $[\mathrm{C}_s, \{t\}] = |\mathrm{C}_s|$.

We claim that without loss of generality, there are no edges in the minimum s-t cut from V_s to C_t; i.e., $[\mathrm{V}_s, \mathrm{C}_t] = 0$. To see this, 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.

So, we have that the minimum s-t cut has value

$$\begin{aligned} \delta c|\mathrm{V}_t| + |\mathrm{C}_s| &\ge \delta c|\mathrm{V}_t| + |N(\mathrm{V}_s)| && \text{(3a)} \\ &\ge \delta c|\mathrm{V}_t| + \delta c|\mathrm{V}_s| \\ &= \delta c|U'| && \text{(3b)} \end{aligned}$$

where (3a) follows from $\mathrm{C}_s \supseteq N(\mathrm{V}_s)$ (since there are no edges from V_s to C_t), and (3b) follows from the expansion of G. □

We note that we have essentially taken the LP dual twice: once in reasoning about a dual witness, and then again by applying the max-flow min-cut theorem. It might be interesting to see a more direct construction of the matching.

Proof of Our Main Theorem

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

Lemma 5: Suppose $|U| \le \frac{\alpha n - 1}{1+\beta}$, where $\beta = \frac{1-\delta}{3\delta - 2}$. Then, we have $|\dot{U}| \le \beta|U|$.

Proof: Assume to the contrary that $|\dot{U}| > \beta|U|$. 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

$$|U + \ddot{U}| = |U| + \lfloor \beta|U| \rfloor + 1 \le |U|(1+\beta) + 1$$

we have $|U + \ddot{U}| \le \alpha n$ by our assumption on $|U|$. Therefore, this set expands, and we have i): $|N(U \cup \ddot{U})| \ge c\delta(|U| + |\ddot{U}|)$.

Furthermore, we have

$$|N(U \cup \ddot{U})| = |N(U)| + |N(\ddot{U}) \setminus N(U)| \le c|U| + |N(\ddot{U}) \setminus N(U)|.$$

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) and using the definition $\lambda = 2(1-\delta) + \frac{1}{c}$, we obtain

$$|\ddot{U}| \leq \frac{(1-\delta)c}{(\delta - \lambda)c + 1}|U| = \beta|U|$$

which is a contradiction. □

We are now ready to prove our main theorem.

Theorem 1: Let $\mathcal{C}$ be a LDPC 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 a $(\alpha n, \delta c)$-expander, where $\delta > 2/3 + 1/(3c)$ and δc is an integer. 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.

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 5, 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, λ)-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 Appendix). 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 [25] in the original paper on expander codes (i.e., decoding success if fewer than $\alpha n/2$ errors using an $(\alpha n, 3c/4)$-expander).

V. Conclusion

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

This paper raises a number of open questions. It would be interesting to see an improvement in the results for LDPC codes. The fraction of error proved here (0.000175, see Appendix) is quite far from the performance of LDPC codes observed in practice. Also, constraining δc to be an integer could require a rather large degree. Both of these problems are a result of the particular method we found for constructing a dual witness, and are not necessarily a deficiency in LP decoding itself. One could improve our results by a more careful weighting scheme, perhaps using graph structures that are more localized than set expansion.

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" [3] LP decoder and memoryless symmetric channel (such as the AWGN channel).

In a follow-up to this work, a subset of the current authors has shown [24], using a dual witness, that LP decoding with expander codes can achieve the capacity of any memoryless symmetric channel in which the bitwise log-likelihood ratio is bounded by some constant. Additionally, a bound is proved for the adversarial channel that is stronger than the one given here. It should be noted, however, that expander codes are much less practical than LDPC codes. Expander codes might have very high—albeit constant—variable degree, and thus are not useful for small finite lengths.

Turbo codes present another promising application of the techniques developed in this paper. Since the distance of turbo codes is in general sublinear [39], one cannot prove a "constant fraction of error" result. However, proving that the WER (of the turbo code LP decoder in [17], [18]) goes to zero as the block length increases (for reasonably high rates) would be a significant result. As far as the authors are aware, no such finite-length error bound is known for turbo codes of any constant rate (other than the results in [17], [22], [21] for the RA (2) cycle code—a code with logarithmic distance).

Appendix
Existence and Construction of Expanders

In this appendix, we give theorems showing that there exist families of expander graphs, and we cite results proving that these expanders can be constructed efficiently. We give these results simply for the sake of completeness of our main result that LDPC codes with LP decoding can correct a constant fraction of errors. The resulting graphs will have large degree requirements and small error-correcting capability, and so improving these results is an important step toward making these codes practical.

A. Expansion From Random Graphs

Using the probabilistic method, one can show the following.

Proposition 6: 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 a $(\alpha n, \delta c)$-expander, where

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

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

- For $i = 1, \ldots, n$ the ith 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

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

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 $s \le n/(2K)^{1/a}$, the quantity $K\left(\frac{s}{n}\right)^a$ is at most $1/2$, and thus we have

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

Thus with probability at least $1/2$, we have that a random graph G formed as described above is a $(\alpha n, \delta c)$-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 6 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 0.000155 fraction of errors.

We note that a more careful analysis of the random bipartite graphs used to prove Proposition 6 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 0.000175 fraction of errors.

To see this, note that the proof of Proposition 6 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 (4). Using a different bound on binomial coefficients we can show that for some $\alpha' > \alpha$ to be described below, all sets 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 $1/2 - o(1)$.

In order to obtain the sharper result, we now apply the following "entropy bound" [40] on the binomial coefficient

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

which is a tighter bound than the previously used $\binom{a}{b} \le \left(\frac{ea}{b}\right)^b$ bound. Using $s = \hat{\alpha} n$, $m = nc/d$ and $t = \delta cs$ this gives

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

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 $0 < \hat{\alpha} \le 0.00035$ causes (6) to be negative. This gives the stronger LP decoding performance bound claimed earlier.

B. 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 [41] through the framework of randomness conductors. Their work implies the following.

Proposition 7: 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 $c = \mathrm{poly}(\log(1-r), 1/(1-\delta))$, 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 $\delta > \frac{1}{c}$ there is some δ' such that $\delta - 1/c \le \delta' \le \delta$ and $\delta' c$ is an integer (note that any $(\alpha n, \delta c)$-expander is clearly also a $(\alpha n, \delta' c)$-expander for any $\delta' \le \delta$). 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.

Acknowledgment

We would like to thank G. David Forney, Jr., David Karger, Ralf Koetter, and Pascal Vontobel for helpful discussions. In addition, we thank the anonymous reviewers for their constructive suggestions that helped to improve the paper.

References

[1] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode linear codes," in *Proc. 37th Annu. Conf. Information Sciences and Systems (CISS'03)*, Baltimore, MD, Mar. 2003.

[2] ——, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.

[3] J. Feldman, D. R. Karger, and M. J. Wainwright, "LP decoding," in *Proc. 41st Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[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 Symp. Theory of Computing*, Montreal, Canada, May 2002, pp. 659–668.

[5] R. Koetter, 2002, personal communication.

[6] R. Koetter and P. O. Vontobel, "Graph-covers and iterative decoding of finite length codes," in *Proc. 3rd Int. Symp. Turbo Codes*, Brest, France, Sep. 2003.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[8] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

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

[10] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[11] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs and belief propagation," in *Proc. 1998 IEEE Int. Symp. Information Theory*, Cambridge, MA, Oct. 1998, p. 117.

[12] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[13] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke, "Finite length analysis of low-density parity check codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[14] N. Wiberg, "Codes and Decoding on General Graphs," Ph.D. dissertation, Linköping Univ., Linköping , Sweden, 1996.

[15] G. D. Forney, Jr., 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*. New York: Springer, 2001, pp. 101–112.

[16] B. Frey, R. Koetter, and A. Vardy, "Signal-space characterization of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 766–781, Feb. 2001.

[17] J. Feldman and D. R. Karger, "Decoding turbo-like codes via linear programming," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, Canada, Nov. 2002.

[18] J. Feldman, "Decoding Error-Correcting Codes via Linear Programming," Ph.D. dissertation, MIT, Cambridge, MA, 2003.

[19] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches," in *Proc. 40th Annu. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 2002.

[20] ——, "Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[21] N. Halabi and G. Even, "Improved bounds on the word error probability of RA(2) codes with linear-programming-based decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 265–280, Jan. 2005.

[22] G. Even and N. Halabi, "Improved bounds on the word error probability of RA(2) codes with linear programming based decoding," in *Proc. 41st Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[23] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 68.

[24] J. Feldman and C. Stein, "LP decoding achieves capacity," in *Proc. Symp. Discrete Algorithms (SODA'05)*, Vancouver, Canada, Jan. 2005.

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

[26] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 782–790, Feb. 2001.

[27] M. Lentmaier, D. V. Truhachev, K. S. Zigangirov, and D. J. Costello, Jr., "An analysis of the block error probability performance of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

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

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

[30] ——, "Error exponents of expander codes under linear-complexity decoding," *SIAM J. Discr. Math.*, vol. 17, no. 3, pp. 426–445, 2004.

[31] ——, "Concatenated codes: Serial and parallel," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1625–1634, May 2005.

[32] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," in *Proc. 34th Annu. Symp. Theory of Computing (STOC)*, Montreal, Canada, May 2002.

[33] R. M. Roth and V. Skachek, "On nearly-MDS expander codes," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun. 2004, p. 8.

[34] V. Guruswami and P. Indyk, "Efficiently decodable low-rate codes meeting the Gilbert Varshamov bound," in *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, Jan. 2004.

[35] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1987.

[36] J. Feldman, D. R. Karger, and M. J. Wainwright, "Linear programmingbased decoding of turbo-like codes and its relation to iterative approaches," in *Proc. 40th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2002.

[37] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

[38] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[39] L. Bazzi, M. Mahdian, S. Mitter, and D. Spielman, "The minimum distance of turbo-like codes," unpublished manuscript, 2001.

[40] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[41] O. Reingold, S. Vadhan, and A. Wigderson, "Entropy waves, the zigzag graph product, and new constant-degree expanders and extractors," in *Proc. 41st IEEE Symp. Foundations of Computer Science (FOCS)*, Redondo, CA, Nov. 2000, pp. 3–13.