LP Decoding Achieves Capacity

Jon Feldman[*]   Cliff Stein[†]

Abstract

We give a linear programming (LP) decoder that achieves the capacity (optimal rate) of a wide range of probabilistic binary communication channels. This is the first such result for LP decoding. More generally, as far as the authors are aware this is the first known polynomial-time capacity-achieving decoder with the maximum-likelihood (ML) certificate *property—where output codewords come with a proof of optimality. Additionally, this result extends the capacity-achieving property of expander codes beyond the binary symmetric channel to a larger family of communication channels.*

Perhaps most importantly, since LP decoding performs well in practice on turbo *codes and* low-density parity-check (LDPC) *codes (comparable to the popular "belief propagation" algorithm), this result exhibits the power of a new, widely applicable "dual witness" technique (Feldman, Malkin, Servedio, Stein and Wainwright, ISIT '04) for bounding decoder performance.*

For expander codes over an adversarial channel, we prove that LP decoding corrects a constant fraction of errors. To show this, we provide a new combinatorial characterization of error events that is of independent interest, and which we expect will lead to further improvements.

1 Introduction

A great deal of current research in coding theory focuses on turbo codes [5], low-density parity-check (LDPC) codes [17] and expander codes [27], using highly efficient "soft information" message-passing (MP) decoders such as belief-propagation and min-sum. These codes and decoders perform[1] very well in practice on probabilistic channels, yet this performance has not been fully explained. The main difficulty lies in the heuristic nature of these message-passing techniques. Operating on graphs that represent the code, the success of an MP decoder depends highly on its "schedule" of updates, which in turn depends on local graph topologies in highly complex ways. These complexities have often been avoided by either (i) assuming an acyclic message schedule, (ii) using bit-flipping decoders that are easier to analyze (but don't perform as well in practice), or (iii) considering simplified noise models or code structures. Each of these assumptions are dissatisfying for various reasons.

Linear programming (LP) decoding is an alternative soft-information decoding algorithm. LP decoders have a simple combinatorial characterization of decoding success, and enjoy a principled technique for bounding performance that avoids the need to make any of the assumptions listed above. LP decoders have been applied successfully to turbo codes and LPDC codes, and in preliminary testing, have been shown to perform as well in practice as message-passing techniques [9]. This performance is not surprising since the error events of the LP and MP decoders are closely related [10, 21, 31, 30], suggesting that, in some sense, the decoding LP represents the problem that MP decoders "want" to solve [22]. The main disadvantage of LP decoding is the complexity that comes from having to solve a linear program. Therefore as a practitioner, one should view LP decoding not necessarily as an alternative decoder to be used in practice, but rather as a principled method to study the performance of modern codes under soft-information decoding. We also note that it is conceivable that LP decoding can be implemented much more efficiently, since the LP's used have only a linear number of highly structured constraints.

Results. We give the first linear-programming (LP) decoders [12] for expander codes. Using a particular family of expander codes due to Barg and Zémor [2], we prove the following performance bounds:

LP decoding achieves the capacity of any binary-input memoryless symmetric LLR-bounded (MSB) channel. (MSB channels include most probabilistic channels commonly considered in practice; see Section 2.1 for a definition). By "achieving capacity," we

[*]Dept. of Industrial Engineering and Operations Research, Columbia University, New York, NY. email: `jonfeld@ieor.columbia.edu`.
Supported by NSF Mathematical Sciences Postdoctoral Research Fellowship DMS-0303407.

[†]Dept. of Industrial Engineering and Operations Research, Columbia University, New York, NY. email: `cliff@ieor.columbia.edu`.
Research partially supported by NSF Grant DMI-9970063.

[1]Throughout the presentation, we will use the verb "perform" in reference to a decoder's ability to correct errors, *not* to its time complexity.

mean that for any rate less than the capacity (optimal rate) of the channel, the probability of decoding error decreases exponentially in the length of the code. This is the first capacity-achieving result for LP decoding.

Under the *adversarial* channel, using an expander code of rate $1 - 2H(\delta)$, the LP decoder corrects an $\delta^2/4 - \epsilon$ fraction of errors, for any $\epsilon > 0$, matching the result of Barg and Zémor [2] using a bit-flipping decoder on the same code. In proving this result, we give a new combinatorial characterization of a necessary condition for LP decoding failure. This characterization is of independent interest, and has parallels to the "pseudocodeword" results for other code families [32, 16, 8, 21].

Our main result shows for the first time that expander codes can achieve capacity for arbitrary MSB channels using a polynomial-time decoder.[2] It was previously known [2, 4, 3] that expander codes can achieve capacity in the binary symmetric channel (BSC), a particular MSB channel.

A *maximum-likelihood (ML) certificate* decoder has the property that any codeword output by the decoder comes with a proof of optimality (maximum-likelihood). Since LP decoders are always ML certificate decoders, we obtain an interesting corollary to our result: *Polynomial-time maximum-likelihood (ML) certificate decoders can achieve capacity.* As far as the authors are aware, this was not known previous to this result.[3]

Techniques and Related Work. LP decoding was introduced in [12] for turbo codes, and has since been extended to LDPC codes [14, 9, 13] and considered in general for binary codes [11, 9]. Our work is the first to consider LP decoding for expander codes, and the LP given here is a natural generalization of the one in [14, 9].

The idea behind LP decoding is to use a linear program to try to find the ML codeword. For a specific code, a polytope $Q \in [0,1]^n$ over variables $\{y_i\}$ is specified such that the integral points in the polytope are exactly the codewords of the code. Using an appropriate objective function, and enforcing $y \in Q$, $y_i \in \{0,1\}$, one obtains an integer linear program that is a maximum-likelihood decoder.

We relax the integer constraints to $0 \le y_i \le 1$ to obtain a LP relaxation that is solvable in polynomial time. Upon solving the LP, if the solution y is integral, then y must represent the ML codeword; if it is fractional, then an error is declared. This gives LP decoders the *ML-certificate* property. The word error rate of an LP decoder is the probability, taken over the noise in the channel, that the transmitted codeword is the optimal solution to the LP.

Our analysis uses the dual of the linear program. We show that coding success is equivalent to the existence of a "dual witness:" a dual feasible solution with objective value 0. Using complimentary slackness, we derive conditions that the dual must satisfy and then explain how to construct such a dual solution. The construction of the dual solution, as a function of the noise in the channel, is one of the key technical ideas.

In recent joint work [13] with Malkin, Servedio and Wainwright, we used a dual witness to prove that LP decoding corrects a constant fraction of errors using LDPC codes. In the current paper, this technique is extended to expander codes, and to both a probabilistic and an adversarial setting. We show that when the code rate is below the capacity of the channel, then with high probability over the noise in the channel, there exists a dual witness proving that the transmitted codeword is optimal. We also introduce new combinatorial conditions that characterize when an expander code fails to decode correctly.

Expander codes were introduced in [27] (see also [29]). One of the original results gave codes of rate $1 - 2H(\delta)$, built on Ramanujan graphs; it was shown that a bit-flipping algorithm (a variant on the algorithm in [17]) corrects a $\delta^2/64$ fraction of errors. Zémor [33] uses another variant of the algorithm to improve this to $\delta^2/4$ (which has since been further improved [28] to $\delta^2/2$).

In later work, Barg and Zémor [2] give a bit-flipping algorithm that achieves the capacity of the binary symmetric channel. Our paper is very much inspired by their work, and our capacity-achieving code construction is roughly equivalent. Using more sophisticated constructions [4, 3], they correct a fraction of

[2]This has also been achieved independently by Roth and Skatchek [25] using a different code and decoder.

[3]We note that our definition of "achieving capacity" requires that the word error rate decrease exponentially in the size of the code. Another definition requires only that the word error rate go to zero as the code size increases; this is achieved trivially by ML decoding a convolutional code with logarithmic distance [22].

errors up to the Zyablov bound, and improve the error probability in the BSC. Guruswami and Indyk [19] give a different expander-based binary code construction, and also attain the Zyablov bound. In later work, they achieve the Gilbert-Varsharmov bound for low rates [20]. It would be interesting to see if LP decoding could improve the results for these constructions.

Density evolution [24, 23] has given near-capacity rate thresholds for distributions of random LDPC codes under BP and min-sum decoding for more general channels, and represents a major breakthrough in the analysis of the message-passing decoders. However the thresholds computed using density evolution are only estimates of the true behavior of the decoders, because they assume a cycle-free message history, which is not the case in practice. That being said, density evolution has yielded thresholds that are quite close to capacity [7]. Burshtein and Miller [6] use expander-based arguments to give further results on message-passing decoders for LDPC codes, incorporating soft-information decoding as well as irregular degree distributions.

Channel capacity for memoryless symmetric channels, under polynomial-time decoding, was first achieved by Forney [15] using concatenated codes and generalized minimum-distance (GMD) decoding.

2 Background

2.1 Coding, channel models. A binary code C of length n is a subset of $\{0,1\}^n$, where $|C| = 2^k$. The code C is used to transmit information in the presence of noise. An information word $x \in \{0,1\}^k$ is encoded to a unique codeword $y \in C$, and sent over a noisy channel. A corrupt codeword $\hat{y}$ is received, and the decoding task is to recover the transmitted codeword y. The *rate* of the code is k/n. The *distance* of the code is the minimum Hamming distance $\Delta(y, y')$ between any two distinct codewords $y, y' \in C$. The *relative distance* is the distance divided by the length of the code. A binary *linear* code $C \subseteq \{0,1\}^n$ is a linear subspace of $\mathbb{F}_2^n$; i.e., $0^n \in C$, and for all codeword pairs $y, y' \in C$, we have $(y + y') \in C$.

We will consider both adversarial and probabilistic noise. In the adversarial model, the channel flips some of the bits arbitrarily. A decoder is said to "correct an α fraction of error" if, for any set of at most αn bits flipped by the channel, the decoder recovers the original codeword. For the probabilistic model, we will consider an arbitrary binary-input *memoryless symmetric LLR-bounded* (MSB) channel, which we now define. Associated with the channel is an alphabet Σ representing the set of possible symbols output by the channel. (Note that this could be a continuous set, such as the reals.) The channel being *memoryless* means that the noise affects each bit transmitted over the channel independently; therefore, the channel is completely specified by transition probabilities $p(\mathtt{a}|\mathtt{b})$ for each $\mathtt{a} \in \Sigma$ and $\mathtt{b} \in \{0,1\}$, where $p(\mathtt{a}|\mathtt{b})$ denotes the probability that symbol $\mathtt{a}$ is output by the channel, given that the bit $\mathtt{b}$ is transmitted. (In continuous alphabets, $p(\mathtt{a}|\mathtt{b})$ is a p.d.f.) The channel being *symmetric* means that the noise affects input 0's and 1's symmetrically; formally, Σ can be partitioned into pairs $(\mathtt{a}, \mathtt{a}')$ where $p(\mathtt{a}|0) = p(\mathtt{a}'|1)$ and $p(\mathtt{a}|1) = p(\mathtt{a}'|0)$. (We also allow for a single "erasure" symbol to be its own pair.) Finally, we define the *log-likelihood ratio* (LLR) γ_i of a received bit $\hat{y}_i$ to be $\gamma_i = \log \frac{p(\hat{y}_i|0)}{p(\hat{y}_i|1)}$. The channel is *LLR-bounded* is there is some number W where $-W < \gamma_i < W$ for all possible received symbols $\hat{y}_i \in \Sigma$.

One common example of an MSB channel is the *binary symmetric channel* (BSC) where each bit is flipped independently with probability p. In the BSC, we have $\Sigma = \{0,1\}$, $\gamma(1|0) = \gamma(0|1) = p$, and $\gamma(1|1) = \gamma(0|0) = 1-p$. We will use the BSC as a running example. An important example of an *unbounded* memoryless symmetric channel is the additive white Gaussian noise (AWGN) channel, where $\Sigma = \mathbb{R}$, and for each transmitted bit y_i, we have $\hat{y}_i = (1 - 2y_i) + \mathcal{N}(0, \sigma^2)$, where $\mathcal{N}(0, \sigma^2)$ is a zero-centered Gaussian with variance σ^2. In practice, we could truncate the tails of the Gaussian, and get an MSB channel.

A decoder is a *Maximum-likelihood* (ML) decoder if it always outputs the codeword y that maximizes the likelihood of receiving $\hat{y}$, given that y was transmitted. Equivalently, it outputs the codeword y that minimizes the quantity $\sum_i \gamma_i y_i$. In the BSC, the ML decoder finds the codeword that is closest in Hamming distance to the received word. The *word error rate* P_{err} of a decoder is the probability, taken over the noise in the channel, that the decoder succeeds (outputs the codeword y that was originally transmitted).

2.2 Channel Capacity. A *code family* is a set of codes of a particular fixed rate r, but increasing length n. A major goal in coding theory is to define a code family (and accompanying decoder) with as high a rate as possible such that $P_{\text{err}} \leq 2^{-\Omega(n)}$. For any memoryless channel, this is achieved by a random code using ML decoding, as long as the rate r is strictly less than the *capacity* $\mathcal{C}$ of the channel [26, 18]. Furthermore, this property is not possible if $r > \mathcal{C}$ (see [18]). The capacity $\mathcal{C}$ is a function only of the channel model (and its associated parameters). For example, the capacity of the binary symmetric channel with crossover probability p is equal to $1-H(p)$, where H is the binary entropy function.

The *random coding exponent* $\mathcal{E}(r)$ is a standard lower bound on the expectation of $-(\log P_{\text{err}})/n$ under ML decoding, taken over a random choice of codes of rate r. This random coding exponent [18] has the property that $\mathcal{E}(r) > 0$ for all rates $r < \mathcal{C}$, and has been studied extensively for different channel models.

2.3 Expander codes. Let $G = (V, E)$ be a d-regular graph with $M = |V|$ nodes and $N = |E|$ edges. For a node $j \in V$, we $\Gamma(j)$ be the set of d edges incident to j. We will choose a graph G that is an expander, but we do not need this property to define the code.

An *expander code* [27] (see also [2]) is a code based on G, defined as follows.[4] For each node $j \in V$, let C_j be a binary linear code with length d, rate r_j, and relative distance δ_j. For each node $j \in V$, define an arbitrary (but fixed) ordering of the edges incident to j. The *expander code* $\mathbf{EC}(G, \{C_j\}_j)$ is defined as the settings of bits y_e to the edges $e \in E$ such that for every node $j \in V$, the bits $\{y_e\}_{e\in\Gamma(j)}$ (when considered in their fixed ordering) form a codeword of C_j. For a codeword $c \in C_j$, and some edge $e \in \Gamma(j)$, let $c[e] \in \{0, 1\}$ be the bit assigned to edge e in the codeword c. By counting the number of linear constraints on the code, it is easily seen that the rate R of the overall expander code C is at least $1 - 2(\sum_j (1 - r_j))/M$.

3 LP decoding with expander codes

In this section we define an LP decoder for an arbitrary expander code. This decoder is a natural generalization of the one for LDPC codes given in [14].

The decoding LP contains a variable $f_e \geq 0$ for every edge in the graph, indicating the value of the code bit y_e. The LP objective is to minimize $\sum_e \gamma_e f_e$, where γ_e is a function of the channel model, and the received word $\hat{y}$. For probabilistic channels, γ_e is defined to be the LLR $\gamma_e = \log \frac{p(\hat{y}_e|0)}{p(\hat{y}_e|1)}$ of the code bit associated with edge e, as discussed in Section 2.1. For adversarial channels, where $\hat{y} \in \{0, 1\}^N$, we set $\gamma_e = +1$ if $\hat{y}_e = 0$, and $\gamma_e = -1$ if $\hat{y}_e = 1$. This makes the LP objective, for all integral solutions $f_e \in \{0, 1\}^N$, equal to $\Delta(f_e, \hat{y}_e) - \sum_e \hat{y}_e$, an adjusted Hamming distance from the received word. We also have auxiliary variables $w_{j,c} \geq 0$ defined for each node j and local codeword $c \in C_j$, indicating that node j is satisfied by the local codeword c. When $w_{j,c} = 1$ it should be the case that the edges $e \in \Gamma(j)$ take on values $f_e = c[e]$. The LP constraints enforce consistency between the f and w variables in the natural way, specified in the LP below:

$$\text{minimize} \quad \sum_e \gamma_e f_e \quad \text{s.t.}$$

$$\forall j \in V, \quad \sum_{c \in C_j} w_{j,c} = 1$$

$$\forall e = (j, j') \in E, \quad f_e = \sum_{\substack{c \in C_j: \\ c[e]=1}} w_{j,c} = \sum_{\substack{c \in C_{j'}: \\ c[e]=1}} w_{j',c}$$

We claim that solutions (f, w) to this LP where $f \in \{0, 1\}^N$ must have $f \in C$. To see this, consider a single node $j \in V$. By the LP constraints, the variables $\{f_e\}_{e\in\Gamma(j)}$ must represent a convex combination of local codewords $c \in C_j$. However, since $f_e \in \{0, 1\}$ for all e, the convex combination must put all its weight on a single local codeword. Therefore, since this holds for all j, we have $f \in C$. This also shows that this LP decoder has the ML-certificate property.

Another property we will need is that the decoding polytope is *C-symmetric* (see [11, 9]). We include the a proof of this fact in the full version. The C-symmetry of the decoding polytope allows us to assume (for the purposes of analysis) that 0^N is the transmitted codeword.

3.1 Bounding the word error rate using a dual witness. In this section we describe the method of proving a word error rate bound using a zero-valued dual feasible point. This method was first described in [13] in order to show that LP decoders correct a constant fraction of error using LDPC codes.

When we assume 0^N is transmitted, our LP decoder succeeds if it outputs a solution where $f = 0^N$. There

[4] We use the definition in [2]; more general expander codes, to which LP decoding can also be applied, can be found in [29].

is only one feasible setting of the $\{w_{j,c}\}$ variables when $f = 0^N$; namely, $w_{j,0^d} = 1$ for all j, and all other $w_{j,c} = 0$. We refer to this setting of the $w_{j,c}$ variables as w^*, and so our LP decoder succeeds if $(0^N, w^*)$ is the unique LP optimum. (If there are multiple LP optima, we assume failure.) The solution $(0^N, w^*)$ is always feasible, and always has value zero. Therefore, a necessary and sufficient condition for $(0^N, w^*)$ to be optimal is the existence of a dual feasible solution with value zero. Furthermore, a sufficient condition for $(0^N, w^*)$ to be the *unique* optimum is the existence of zero-valued dual feasible solution with slack in every dual constraint associated with variables f_e. (This follows from complementary slackness.) Our strategy for proving decoding success will be to find such a dual solution.

If we take the LP dual, set the objective value equal to zero, enforce slack in the edge constraints, and simplify, we get the following (open) polytope, defined over variables $\{\tau_{e,j}\}_{j \in V, e \in \Gamma(j)}$:

$$\hat{P}: \quad \forall j \in V, c \in C_j, \quad \sum_{e \in \Gamma(j)} c[e] \cdot \tau_{e,j} \geq 0 \tag{3.1}$$

$$\forall e = (j, j') \in E, \quad \tau_{e,j} + \tau_{e,j'} < \gamma_e \tag{3.2}$$

Any feasible point $\tau \in \hat{P}$ is a proof of the fact that 0^N is the unique primal LP optimum.

As an sanity check, suppose we transmitted over an adversarial channel, and there were no errors; i.e., $\gamma_e = +1$ for all $e \in E$. Then, if we set all $\tau_{e,j} = 0$, we get a point in $\hat{P}$. Thus if there are no errors, the decoder succeeds. In later sections, we will demonstrate feasible points in $\hat{P}$ for more interesting situations.

4 Graph expansion and ρ-orientations

In this section we state some graph-theoretic definitions and lemmas that we need to construct points in the polytope $\hat{P}$, for both probabilistic and adversarial error. Proofs are given in the full version.

DEFINITION 4.1. *A d-regular graph G is a (α, ρ)-*expander *if, for all vertex–induced subgraphs $G' = (V', E')$ with $|V'| \leq \alpha |V|$, we have $|E'| \leq \rho d |V'|$.*

LEMMA 4.1. *In a d-regular (α, ρ)-expander $G = (V, E)$, if some subgraph $G' = (V', E')$ has $|E'| \leq \alpha \rho d |V|$, then $|V'| \geq |E'|/(\rho d)$.*

DEFINITION 4.2. *Let a ρ-*orientation *of a subgraph $G' = (V', E')$ of a d-regular graph G be an assignment of directions to every edge in E' such that each node in V' contains at most ρd incoming edges from E'.*

LEMMA 4.2. *If a d-regular graph G is a (α, ρ)-expander, where ρd is an integer, then all subgraphs $G' = (V', E')$ where $|E'| \leq \alpha \rho d |V|$ contain a ρ-orientation.*

THEOREM 4.1. *(Alon-Chung[1]) Let $G = (V, E)$ be a d-regular graph such that all eigenvalues other than d have absolute value at most λ. Let T be a subset of the vertices of G of size $\gamma |V|$. Then, the number of edges contained in the subgraph induced by T in G is at most $\gamma |V| \left(\frac{d\gamma}{2} + \frac{\lambda}{2}(1 - \gamma) \right)$.*

For a particular value ρ, we can use Theorem 4.1 to construct a d-regular graph that is an (α, ρ)-expander, where $\alpha = 2\rho - (\lambda/d)$.

5 Probabilistic error: achieving capacity

In this section we assume an arbitrary MSB channel with capacity $\mathcal{C}$ and LLR bound W. Our task is to come up with an expander code family of some given rate $R < \mathcal{C}$ such that the word error rate under LP decoding decreases exponentially in the code length $N = |E|$.

5.1 The parameters of the code. The expander code family we present here is essentially the same as that of Barg and Zémor [2], with some of the parameters set differently. We let G be a balanced bipartite d-regular Ramanujan graph with second-largest eigenvalue $\lambda = \Theta(\sqrt{d})$, as used in [2]. Since G is bipartite, we have $V = \{A, B\}$, with $|A| = |B| = M/2$. We use two codes C_A and C_B, and set $C_j = C_A$ for all $j \in A$, and $C_j = C_B$ for all $j \in B$. Let R be some target rate of our overall code. We let r_{A}, the rate of code C_A, be any rate greater than R, and set r_{B}, the rate of the code C_B, to be equal to $r_{\text{B}} = R - r_{\text{A}} + 1$. Note that since $r_{\text{A}} > R$, we have $r_{\text{B}} < 1$. The overall code will have rate at least $1 - 2(\sum_j r_j)/M = r_{\text{A}} + r_{\text{B}} - 1 = R$, as required.

Let δ_{A} and δ_{B} be the relative distance of the code C_A and C_B, respectively. We use the following definition to further characterize the code C_A:

DEFINITION 5.1. *For a particular memoryless symmetric channel, a binary linear code C of length n is*

(β, κ)-robust *if, with probability at least* $1 - 2^{-\kappa n+1}$ *over the noise in channel, all non-zero codewords* $y \in C$ *have cost* $\sum_i \gamma_i y_i \geq \beta n$.

For now we assume that C_A is (β, κ)-robust for some $\beta, \kappa > 0$. We will show later that this can be achieved for rates r_{A} less than capacity. We define $\rho' = \delta_{\mathrm{B}}/(1 + \delta_{\mathrm{B}}/\delta_{\mathrm{A}} + W/\beta)$. We let ρ be any number where ρd is an integer, and $\rho'/2 \leq \rho \leq \rho'$. By Theorem 4.1, we can make G a (α, ρ)-expander, where $\alpha = 2\rho - (\lambda/d)$. (Note that we may need to increase d in order to define ρ, and to make $\alpha > 0$.)

We use the notation C_{prob} to represent a particular expander code $\mathbf{EC}(G, \{C_A\}_{j\in A}; \{C_B\}_{j\in B})$ as described above. (The specific parameters will be clear from context.)

5.2 Finding a point in $\hat{P}$. We use the LP decoder from Section 3 on the code C_{prob}, and this defines a polytope $\hat{P}$. Our goal is to construct a point in $\hat{P}$, as long as some high-probability event occurs. Such a point is a dual witness to the optimality of 0^n in the primal decoding LP, and therefore a proof that the decoder succeeds.

We assume a particular received word $\hat{y}$, and the resulting edge costs $\gamma_e = \log \frac{p(\hat{y}_e|0)}{p(\hat{y}_e|1)}$, where $-W < \gamma_e < W$. The *cost* $\gamma(c)$ of a local codeword $c \in C_j$ for some node j is equal to $\gamma(c) = \sum_{e\in\Gamma(j)} c[e]\gamma_e$.

Suppose we have that for every $j \in A$, all non-zero codewords $c \in C_j$ have positive cost. In this case, finding a point in $\hat{P}$ is simple: just set $\tau_{e,j} = \gamma_e - \epsilon$ for all $j \in A$, and set $\tau_{e,j} = 0$ for all $j \in B$, for some small $\epsilon > 0$. When some non-zero codewords have non-positive cost, we need to be more careful about how we set the variables $\tau_{e,j}$, which we will refer to as "edge weights." If a node has a negative-cost local codeword, then we need to bias the incident edge weights to be positive in order to satisfy the node constraints (3.1) of $\hat{P}$; on the other hand, if a node has all its non-zero codewords with positive cost, then it can afford to "absorb" some incident excess negative weight.

This motivates the following definition: let $T \subseteq A$ be the nodes in A that have an incident non-zero local codeword with cost less than or equal to βd. Formally, $T = \{j \in A : \exists c \in C_j \text{ s.t. } \sum_{e\in\Gamma(j)} c[e] \cdot \gamma_e \leq \beta d\}$. These are the "bad" nodes in A, the ones that cannot afford to absorb positive weight, and therefore must be treated carefully. Note that since we made the code C_A robust, it will be unlikely for a node in A to be bad.

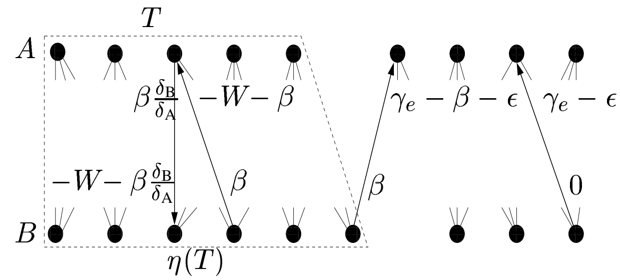

Figure 1: Setting the edge weights $\tau_{e,j}$ for each node j and incident edge $e \in \Gamma(j)$ to satisfy the constraints of $\hat{P}$. The weights are set according to the "bad" nodes T, their neighbors $\eta(T)$, and the orientation of each edge.

Let $\eta(T)$ be nodes in the neighborhood of T; note that $\eta(T) \subseteq B$, since the graph is bipartite. Our weighting scheme is given in the proof of the following theorem (also see Figure 1):

THEOREM 5.1. *If* $|T \cup \eta(T)| \leq \alpha M$*, then the LP decoder succeeds (outputs the transmitted codeword).*

Proof. We show the LP decoder succeeds by providing a point in $\hat{P}$. To set the edge weights $\tau_{e,j}$, we first define a direction for each edge in the graph. All edges that are not incident to T are directed toward the nodes A. Edges incident to T are directed according to a ρ-orientation of the subgraph induced by $(T \cup \eta(T))$. This is possible using Theorem 4.2, since $|T \cup \eta(T)| \leq \alpha M$ by assumption, and so $|\{\Gamma(j)\}_{j\in T}| \leq \alpha\rho d M$ by expansion.

We will give each edge $e = (j \to j')$ a "tail-weight" $\tau_{e,j}$ and a "head-weight" $\tau_{e,j'}$. To satisfy the edge constraints (3.2) of $\hat{P}$, the sum of these two weights should be strictly less than γ_e. We give the assignment in detail below (also in Figure 1), where $\epsilon > 0$ is a small constant to be specified later:

(i) For all edges leaving T, set the tail-weight to $\beta(\delta_{\mathrm{B}}/\delta_{\mathrm{A}})$ and the head-weight to $-W - \beta(\delta_{\mathrm{B}}/\delta_{\mathrm{A}})$. Note that the sum is $-W$, which is strictly less than γ_e by definition of W.

(ii) For all edges going into T, set the head-weight to $-W - \beta$ and the tail-weight to β, and again the sum is $-W < \gamma_e$.

(iii) For all edges e incident to $\eta(T)$ but not T, set the tail weight to β, and the head weight to $\gamma_e - \beta - \epsilon$. Note that these edges are all directed away from $\eta(T)$. The sum of the edge weights is $\gamma_e - \epsilon < \gamma_e$.

(iv) For all other edges (those not incident to either T or $\eta(T)$), set the head-weight to $\gamma_e - \epsilon$ and the tail-weight to 0. Recall that these edges are all directed toward A. The sum of these two edge weights is also $\gamma_e - \epsilon < \gamma_e$.

We show that this weight assignment satisfies the node constraints (3.1) of $\hat{P}$ using three cases:

(i) For a node $j \in T$, we have at most $\rho d \leq \rho' d$ incoming edges e with weight $\tau_{e,j} = -W - \beta$; the remaining (outgoing) edges have weight $\beta(\delta_{\text{B}}/\delta_{\text{A}})$. Each non-zero codeword $c \in C_j$ has a support set of size at least $\delta_{\text{A}} d$, and so for all $c \in C_j$ we have $\sum_{e \in \Gamma(j)} c[e] \cdot \tau_{e,j} \geq \rho' d(-W - \beta) + (\delta_{\text{A}} - \rho') d\beta(\delta_{\text{B}}/\delta_{\text{A}}) = 0$.

(ii) For a node $j \in \eta(T)$, we have at most $\rho d \leq \rho' d$ incoming edges e with weight $\tau_{e,j} = -W - \beta(\delta_{\text{B}}/\delta_{\text{A}})$, and the remaining (outgoing) edges have weight β. Therefore, similar to the previous case, every non-zero codeword $c \in C_j$ has $\sum_{e \in \Gamma(j)} c[e] \cdot \tau_{e,j} \geq \rho' d(-W - \beta(\delta_{\text{B}}/\delta_{\text{A}})) + (\delta_{\text{B}} - \rho') d\beta = 0$.

(iii) For a node $j \in (A - T)$, every incident edge e is incoming, and has weight $\tau_{e,j}$ equal to either $\gamma_e - \epsilon$ or $\gamma_e - \beta - \epsilon$. In the worst case and wlog, they all have weight $\tau_{e,j} = \gamma_e - \beta - \epsilon$, and so every non-zero codeword $c \in C_j$ has $\sum_{e \in \Gamma(j)} c[e] \cdot \tau_{e,j} \geq (\sum_{e \in \Gamma(j)} c[e] \cdot \gamma_e) - d(\beta + \epsilon)$. By the definition of T, we have $\sum_{e:c[e]=1} \gamma_e > \beta d$, and so there exists $\epsilon > 0$ such that $\sum_{e:c[e]=1} \tau_{e,j} \geq 0$.

(iv) For a node $j \in (B - \eta(T))$, we have $\tau_{e,j} = 0$ for all $e \in \Gamma(j)$. □

The main theorem of the section says that if C_A is robust, then the word error rate of LP decoding decreases exponentially in $N = |E|$. In the next section we use this theorem to attain capacity.

THEOREM 5.2. *If the code C_A is (β, κ)-robust, for some $\beta, \kappa > 0$, then there exists a sufficiently large degree d such that the word error rate of LP decoding using the code C_{prob} is at most $2^{-\Omega(N)}$.*

Proof. By Theorem 5.1 the LP decoder succeeds as long as $|T \cup \eta(T)| \leq \alpha M$. Using $|\eta(T)| \leq d|T|$, we have success if $|T| \leq \alpha M/(d+1)$, which is equivalent to $|T| \leq (M/2)(2\alpha/(d+1))$, i.e., if the fraction of bad nodes in A is at most $\alpha_2 = \frac{2\alpha}{d+1} \geq \frac{2\rho'}{d+1} - \frac{2\lambda}{d(d+1)}$. Since the code C_A is (β, κ)-robust, we have that the probability of a node being bad is at most $2^{-\kappa d+1}$, and so the expected fraction of bad nodes is at most $2^{-\kappa d+1}$. Note that α_2 decreases linearly in d, whereas the expected fraction of bad nodes decreases exponentially in d. Thus, for sufficiently large d, we have $2^{-\kappa d+1} < \alpha_2$. Each node in A is bad independently, since the edges adjacent to them are disjoint. A Chernoff bound implies a word error rate of at most $2^{-\Omega(M)} = 2^{-\Omega(N)}$, since d is constant. □

We note that for the binary symmetric channel, the error exponent (the constant in the Ω) is not as good as the one proved by Barg and Zémor [2] using a bit-flipping decoder; in particular, it has an unfortunate inverse dependence on d. It would be interesting to see if a different method of setting the edge weights could yield stronger results; since LP decoding performs better than bit-flipping decoders (at least on LDPC codes), one would expect this to be possible.

5.3 Achieving capacity. In this section we will need the following theorem, proved in the full version, which is in essence a slight generalization of Shannon's noisy coding theorem:

LEMMA 5.1. *For any memoryless symmetric channel with capacity $\mathcal{C}$, for sufficiently large n, any rate $r < \mathcal{C}$, and any β where $0 < \beta < \mathcal{C} - r$, there exists a $(\beta, \mathcal{E}(r + \beta))$-robust binary linear code C with length n, rate r, and minimum distance at least $H^{-1}(1 - r)$, where $\mathcal{E}$ is the random coding exponent.*

We now define the code C_{cap} that will achieve capacity. We use a particular case of the code C_{prob}. We set r_{A} to some number where $R < r_{\text{A}} < \mathcal{C}$, and β to some number where $0 < \beta < \mathcal{C} - r_{\text{A}}$. We then invoke Lemma 5.1 above (with $r = r_A$) to obtain the code C_A. Thus, the code C_A is $(\beta, \mathcal{E}(r_A + \beta))$-robust. Note that the random coding exponent $\mathcal{E}(r_A + \beta) > 0$ since $r_A + \beta < \mathcal{C}$. We also have $\delta_{\text{A}} = H^{-1}(1 - r_{\text{A}})$. Furthermore we make $\delta_{\text{B}} = H^{-1}(1 - r_{\text{B}})$ by using a code C_B on the Gilbert-Varsharmov bound (see [18]).

We note that any constants δ_{A} and δ_{B} would suffice to achieve capacity; the fact that the codes are on the GV bound only affects the error exponent (the constant in front of N in the exponent). In theory, we use exhaustive search to construct the codes C_A and C_B, which takes constant time, since d is constant. (In practice, note that any codes C_A and C_B with decent parameters give exponentially small word error rate for rates close to capacity, just by using Theorem 5.2.) Theorem 5.2 gives the following:

THEOREM 5.3. *The word error rate of LP decoding using code C_{cap} is at most $2^{-\Omega(N)}$ for all rates $R < \mathcal{C}$.*

6 Adversarial error

A dual witness can also be used to give bounds for the adversarial channel. In the probabilistic channel we gave a condition on the error pattern that implied a dual witness, and then proved that this condition was likely to hold. In the adversarial channel, we give a dual witness assuming a bound on the number of bits flipped by the channel. Specifically, we will show that LP decoding succeeds if $\Delta(y, \hat{y}) \leq \alpha N$, where y is the transmitted codeword, $\hat{y} \in \{0,1\}^n$ is the received word, and α is as high a fraction as possible. In this section, our edge costs γ_e are defined so that the LP minimizes the Hamming distance from the received word $\hat{y}$, as explained in Section 3: we set $\gamma_e = +1$ if $\hat{y}_e = 0$, and $\gamma_e = -1$ if $\hat{y}_e = 1$, and so we have $\sum_e \gamma_e y'_e = \Delta(y', \hat{y}) - \sum_e \hat{y}_e$ for all codewords $y' \in C$.

To prove that LP decoding succeeds, we find a point in the polytope $\hat{P}$, as in the previous section. (We may assume that 0^N is transmitted, since the LP is C-symmetric, which we prove in the full version.) We first show, in Section 6.1, a general result for an arbitrary expander code, giving a purely combinatorial necessary condition for the LP decoder to fail. We follow this up in Sections 6.2 and 6.3 with a specific construction of an expander code that takes advantage of this condition.

6.1 Necessary combinatorial failure condition: error cores. We define $C_{\text{core}} = \mathbf{EC}(G, \{C_j\})$ to be an arbitrary expander code built on a d-regular graph G where each code C_j has relative distance at least δ. Note that C_{core} assumes nothing about the expansion of the graph. The following combinatorial object will be key to our results in this section:

DEFINITION 6.1. *A* ρ-error core *is a subgraph* $G' = (V', E')$ *where* (i) $\hat{y}_e = 1$ *for all* $e \in E'$*, and* (ii) $\Gamma(j) \cap E' \geq \rho d$ *for all* $j \in V'$*.*

For an edge e to be in an error core, the code bit y_e must be flipped by the channel, and both endpoints of e must be incident to at least ρd edges e' that are also in the error core. This can become quite restrictive. We now state the main theorem in this section, which will later lead to a bound on the adversarial channel.

THEOREM 6.1. *If the LP decoder fails in the adversarial channel using code C_{core}, then there exists an $(\delta/4)$-error core in the graph G.*

This theorem should be of independent interest, since it does not rely on graph expansion; it is merely a graph-theoretic necessary condition for decoding failure. This type of characterization is often referred to as a "pseudocodeword," since it is an object that "fools" a sub-optimal decoder. (For example, the "stopping sets" of an LDPC code represent pseudocodewords for belief-propagation in the binary erasure channel [8].)

The rest of this section is devoted to proving Theorem 6.1. For some received vector $\hat{y}$, let S^0 be the set of edges with an error; i.e., $S^0 = \{e \in E : \gamma_e = -1\}$. Define sets $S^1 \supseteq S^2 \supseteq \ldots$ and $T^1 \supseteq T^2 \supseteq \ldots$ inductively as follows: Let $T^i \subseteq V$ be the set of nodes with at least $(\delta/4)d$ incident edges in S^{i-1}. Now define $S^i \subseteq S^{i-1}$ to be the set of edges in S^{i-1} induced by T^i. Note that this definition could produce an infinite sequence of sets (e.g., if $S^0 = E$).

LEMMA 6.1. *If $S^i = \emptyset$ for some finite i, the LP decoder succeeds.*

Proof. We show decoding success by constructing a point in $\hat{P}$. We set edge weights $\tau_{e,j}$ as follows, where $\epsilon > 0$ is a small constant that we specify later:

(i) For all $e = (j, j') \notin S^0$: set $\tau_{e,j} = \tau_{e,j'} = 1/2 - \epsilon$. Since $e \notin S^0$, we have $\gamma_e = +1$, and so $\tau_{e,j} + \tau_{e,j'} = 1 - 2\epsilon < \gamma_e$.

(ii) For all i, and edges $e = (j, j') \in S^i$ but not in S^{i+1}: By definition of T^{i+1}, at most one endpoint of e is in T^{i+1}. If neither endpoint is in T^{i+1}, set the two weights $\tau_{e,j}$ and $\tau_{e,j'}$ to $1/2 - \epsilon$ and $-3/2$ arbitrarily. If one endpoint (say j) is in T^{i+1}, set $\tau_{e,j} = 1/2 - \epsilon$ for that endpoint, and $\tau_{e,j'} = -3/2$ for the other endpoint. In both cases, we have $\tau_{e,j} + \tau_{e,j'} = -1 - \epsilon < -1 = \gamma_e$.

Since $S^i = \emptyset$ for some finite i, all edges fall into one of the two cases above. We claim that τ is a feasible point in $\hat{P}$. We have already argued that the edge constraints (3.2) of $\hat{P}$ are satisfied, and so it remains to show that the node constraints (3.1) are satisfied.

We first show that every node j has fewer than $(\delta/4)d$ incident edges e with $\tau_{e,j} = -3/2$. (i) Consider a node $j \notin T^1$. This node is incident to fewer than $(\delta/4)d$ edges in S^0, and these are the only edges e that could possibly have $\tau_{e,j} = -3/2$. (ii) Consider a node $j \in T^1$. Since $S^i = \emptyset$ for some i, we have $j \in (T^i - T^{i+1})$ for some i. Since $j \notin T^{i+1}$, there are fewer than $(\delta/4)d$ edges in $\Gamma(j) \cap S^i$. If some edge

$e \notin S^i$, then $\tau_{e,j} = 1/2 - \epsilon$. Therefore, fewer than $(\delta/4)d$ incident edges have $\tau_{e,j} = -3/2$.

Thus, there is some $\epsilon' > 0$ such that every node j has at most $(\delta/4 - \epsilon')d$ edges $e \in \Gamma(j)$ with $\tau_{e,j} = -3/2$. Since code C_j has relative distance δ, we have, for all j and $c \in C_j$, $\sum_{e\in\Gamma(j)} c[e]\cdot\tau_{e,j} \geq (-3/2)(\delta/4-\epsilon')d + (1/2-\epsilon)(3\delta/4)d = (3\epsilon'/2 - 3\delta\epsilon/4)d$. Setting $\epsilon \leq 2\epsilon'/\delta$, we get $\sum_{e\in\Gamma(j)} c[e]\cdot\tau_{e,j} \geq 0$. □

LEMMA 6.2. *If there is no $(\delta/4)$-error core in the graph G, then $S^i = \emptyset$ for some finite i.*

Proof. Suppose there is no finite i where $S^i = \emptyset$. Then, for some i, $S^i = S^{i+1} \neq \emptyset$, and so $T^i = T^{i+1}$. This implies, by definition of T^{i+1}, that every node in T^{i+1} has at least $(\delta/4)d$ incident edges in $S^i = S^{i+1}$. Since the edges S^{i+1} are all induced by T^{i+1}, and $S^{i+1} \subseteq S^0$, we have that (T^{i+1}, S^{i+1}) is a $(\delta/4)$-error core. □

Theorem 6.1 follows from Lemmas 6.1 and 6.2.

6.2 Using expansion in the error core. Even if the graph contains an error core, it may be possible to assigning legal edge weights. If the graph expands, then we can use a ρ-orientation to assign the edge weights.

THEOREM 6.2. *Suppose $G = (V,E)$ is a $(\alpha, \delta/4-\epsilon')$-expander, for some $\epsilon' > 0$ where $d(\delta/4-\epsilon')$ is an integer. Then, if the LP decoder fails, there exists a $(\delta/4)$-error core $G' = (V', E')$ where $|E'| > \alpha(\delta/4-\epsilon')dM$.*

Proof. (Sketch) If the LP decoder fails, then by Theorem 6.1 we have an $(\delta/4)$-error core $G' = (V', E')$ in the graph G. Suppose $|E'| \leq \alpha(\delta/4-\epsilon')dM$. We show decoding success by constructing a point in $\hat{P}$. For the edges not in E', set the edge weights as in the proof of Theorem 6.1, using some value $\epsilon > 0$ that we specify later. Since G is a $(\alpha, \delta/4-\epsilon')$-expander, $d(\delta/4-\epsilon')$ is an integer, and $|E'| \leq \alpha(\delta/4-\epsilon')dM$, there exists a $(\delta/4-\epsilon')$-orientation of G' (by Theorem 4.2). Set weights of edges $(j \to j') \in E'$ according to this orientation by setting $\tau_{e,j} = 1/2-\epsilon$ and $\tau_{e,j'} = -3/2$.

This setting of the edge weights clearly satisfies the edge constraints of $\hat{P}$. Also, using the argument from Theorem 6.1 and the ρ-orientation definition, we have that each node j has fewer than $(\delta/4-\epsilon'')d$ incident edges e with weight $\tau_{e,j} = -3/2$, for some $\epsilon'' > 0$. Setting $\epsilon \leq 2\epsilon''/\delta$, we have $\sum_{e\in\Gamma(j)} c[e]\cdot\tau_{e,j} \geq 0$ for all nodes j and non-zero codewords $c \in C_j$. □

6.3 Correcting a $\delta^2/4$ fraction of errors. In order to use Theorem 6.2, we define an expander code $C_{\text{adv}} = \mathbf{EC}(G, \{C\}_j)$ using any d-regular Ramanujan graph G with second-largest eigenvalue $\lambda = \Theta(\sqrt{d})$. The codes C_j for each node j will be identical codes C on the G-V bound, of length d and relative minimum distance δ. (We make d sufficiently large to reach the G-V bound.) The overall code C_{adv} thus has rate $R = 1 - 2H(\delta)$.

THEOREM 6.3. *For any $\epsilon > 0$, there exists a sufficiently large degree d such that using the code C_{adv}, LP decoding corrects a $\delta^2/4 - \epsilon$ fraction of errors in an adversarial channel.*

Proof. (Sketch) As before, using Theorem 4.1, we have that G is a (α,ρ)-expander, where $\alpha = 2\rho - (\lambda/d)$. Setting $\rho = (\delta/4-\epsilon')$ (we later specify $\epsilon' > 0$ s.t. $d(\delta/4-\epsilon')$ is an integer), we get $\alpha = \delta/2 - 2\epsilon' - (\lambda/d)$. By Theorem 6.2, for the LP decoder to fail, there must be an error core with more than $\alpha(\delta/4-\epsilon')dM$ edges. All edges in an error core represent errors, and so there must have been at least $\alpha(\delta/4-\epsilon')dM = (\delta - 4\epsilon' - 2\lambda/d)(\delta/4-\epsilon')N$ errors in the channel. This can be made greater than $(\delta^2/4-\epsilon)N$ by increasing d and decreasing ϵ', maintaining $d(\delta/4-\epsilon')$ an integer. □

7 Conclusions and Future work

We have showed that LP decoding is a strong enough technique to achieve the capacity of an arbitrary MSB channel by using expander codes. However, we still have a lot to learn about the impressive empirical performance of more practical codes like turbo codes and LDPC codes. Since LP decoders apply to these codes (see [9]), we should be able use the techniques developed here to answer the following open questions.

(1) Can we prove capacity (or near capacity) results for LP decoding on LDPC codes? (2) Can we achieve capacity with LP decoding with complexity that does not depend exponentially on the gap to capacity? (3) Can we give a turbo-like code where LP decoding has a word error rate of $2^{-\Omega(n^\epsilon)}$ $(0 < \epsilon < 1)$, for rates close to capacity?

Acknowledgments

We gratefully acknowledge Rocco Servedio, Tal Malkin, Ralf Koetter, Pascal Vontobel, Alexander Barg and Gilles Zémor for helpful discussions, and the reviewers for pointing us to [6].

References

[1] N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. *Disc. Math*, 72, 1988.

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

[3] A. Barg and G. Zémor. Concatenated codes: Serial and parallel. Manuscript, submitted to IEEE Trans. on Information Theory, 2003.

[4] A. Barg and G. Zémor. Error exponents of expander codes under linear-complexity decoding. *SIAM Journal on Discrete Math*, 17(3):426–445, 2004.

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

[6] D. Burshtein and G. Miller. Expander graph arguments for message-passing algorithms. *IEEE Trans. on Information Theory*, pages 782–790, February 2002.

[7] S.-Y. Chung, G. D. Forney, T. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Communications Letters*, 5(2):58–60, February 2001.

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

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

[10] J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo-like codes and its relation to iterative approaches. In *Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing*, October 2002.

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

[12] 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.

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

[15] G. D. Forney. *Concatenated Codes*. M.I.T., 1966.

[16] 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.

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

[18] R. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, New York, NY, 1968.

[19] V. Guruswami and P. Indyk. Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In *Proc. of the 34th annual Symp. on Theory of Computing (STOC)*, 2002.

[20] V. Guruswami and P. Indyk. Efficiently decodable low-rate codes meeting the Gilbert Varshamov bound. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004.

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

[22] Ralf Koetter. Personal communication, 2004.

[23] 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.

[24] T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. on Info. Theory*, 47(2), Feb. 2001.

[25] R. M. Roth and V. Skachek. On nearly-MDS expander codes. In *International Symposium on Information Theory (ISIT '04), Chicago, IL*, June 2004.

[26] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. Journal*, 27:379–423, 1948.

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

[28] V. Skachek and R. Roth. Generalized minimum distance iterative decoding of expander codes. In *Proc. IEEE Information Theory Workshop*, 2003.

[29] D. Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*. PhD thesis, Massachusetts Institute of Technology, 1995.

[30] P. O. Vontobel and R. Koetter. Lower bounds on the minimum pseudo-weight of linear codes. In *International Symposium on Information Theory (ISIT '04), Chicago, IL*, June 2004.

[31] P. O. Vontobel and R. Koetter. On the relationship between linear programming decoding and max-product decoding. Manuscript, submitted to ISITA 2004, Parma, Italy, May 2004.

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

[33] G. Zémor. On expander codes. *IEEE Trans. on Information Theory*, 47(2):835–837, 2001.